



HAL
open science

Artificial intelligence techniques to improve quality of transmission estimation in optical networks

Reda Ayassi

► **To cite this version:**

Reda Ayassi. Artificial intelligence techniques to improve quality of transmission estimation in optical networks. Machine Learning [stat.ML]. Institut Polytechnique de Paris, 2023. English. NNT : 2023IPPAS016 . tel-04521259

HAL Id: tel-04521259

<https://theses.hal.science/tel-04521259>

Submitted on 26 Mar 2024

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Artificial Intelligence Techniques to Improve Quality of Transmission Estimation in Optical Networks

Thèse de doctorat de l'Institut Polytechnique de Paris
Préparée à Telecom SudParis

École doctorale n° 626 de l'Institut Polytechnique de Paris (EDIPP)
Spécialité de doctorat: Mathématiques et Informatique

Thèse présentée et soutenue à Évry-Courcouronnes, le 12/12/2023, par

Reda Ayassi

Composition du Jury :

Catherine Lepers Professeur, Institut-Mines Telecom, Telecom SudParis - France	Président
Massimo Tornatore Professeur, Polytechnique de Milan - Italie	Rapporteur
John Puentes Professeur, IMT Atlantique - France	Rapporteur
Yvan Pointurier Ingénieur de recherche, Huawei - France	Examineur
Michel Morvan Maître de conférence, IMT Atlantique - France	Examineur
Noel Crespi Professeur, Institut-Mines Telecom, Telecom SudParis - France	Directeur de thèse
Ahmed Triki Ingénieur de recherche, Orange Innovation - France	Co-Directeur de thèse
Maxime Laye Data scientist, Orange Innovation – France	Invité
Roberto Minerva Maître de conférence, Mines Telecom, Telecom SudParis - France	Invité

Doctor of Philosophy (PhD) Thesis
Institut-Mines Télécom, Télécom SudParis
& Institut Polytechnique de Paris (IP Paris)

Specialization

COMPUTING, DATA AND ARTIFICIAL INTELLIGENCE

presented by

Reda Ayassi

<p>Artificial Intelligence Techniques to Improve Quality of Transmission Estimation in Optical Networks</p>
--

Commitee:

Massimo Tornatore	Reviewer	Professor, Politecnico di Milano - Italy
John Puentes	Reviewer	Professor, IMT Atlantique - France
Yvan Pointurier	Examiner	Research Engineer, Huawei - France
Michel Morvan	Examiner	Associate Professor, IMT Atlantique - France
Catherine Lepers	Examiner	Professor, Institut-Mines Telecom, Telecom SudParis - France
Ahmed Triki	Advisor	Research Engineer, Orange Innovation - France
Noel Crespi	Advisor	Professor, Institut-Mines Telecom, Telecom SudParis - France
Maxime Laye	Guest	Data Scientist, Orange Innovation - France
Roberto Minerva	Guest	Adjunct Professor, Institut-Mines Telecom, Telecom SudParis - France

**Thèse de Doctorat (PhD) de
Institut-Mines Télécom, Télécom SudParis
et l'Institut Polytechnique de Paris (IP Paris)**

Spécialité

INFORMATIQUE, DONNÉES ET INTELLIGENCE ARTIFICIELLE

présentée par

Reda Ayassi

Techniques Intelligence Artificielle pour l'Amélioration de l'Estimation de la Qualité de Transmission dans les Réseaux Optiques

Jury composé de :

Massimo Tornatore	Rapporteur	Professeur, Polytechnique de Milan - Italie
John Puentes	Rapporteur	Professeur, IMT Atlantique - France
Yvan Pointurier	Examineur	Ingénieur de recherche, Huawei - France
Michel Morvan	Examineur	Maître de conférence, IMT Atlantique - France
Catherine Lepers	Examineur	Professeur, Institut-Mines Telecom, Telecom SudParis - France
Ahmed Triki	Encadrant	Ingénieur de recherche, Orange Innovation - France
Noel Crespi	Directeur de thèse	Professeur, Institut-Mines Telecom, Telecom SudParis - France
Maxime Laye	Invité	Data scientist , Orange Innovation - France
Roberto Minerva	Invité	Maître de conférence, Institut-Mines Telecom, Telecom SudParis - France

Dedication

To My Family and Friends Who Have Always Believed in Me. Thank you all.

Acknowledgements

First and foremost, I would like to thank Ahmed Triki, M. Maxime Laye and the Orange Innovation team who gave me the opportunity to do this research work, welcomed me into the team and provided me all the guidance and support that I needed. The three years I spent at Orange were very fulfilling and educational. I am particularly grateful for Ahmed for his continuous support and advice. He went above and beyond his responsibility and this thesis would not be possible without his aid.

Then I would like to thank my thesis director Prof. Noel Crespi and co-supervisor Dr. Roberto Minerva who have supported me throughout my research.

Finally I would like to thank my family and friends for providing emotional and mental support during the years of my PhD.

Reda Ayassi
4th October 2023

Abstract

Estimating lightpath Quality of Transmission (QoT) is crucial in network design and service provisioning. Recent studies have turned to artificial intelligence (AI) techniques to improve the accuracy of QoT estimation using the data generated by the operational network. We distinguish four categories of solutions. The first category consists of building AI models to check the feasibility of a lightpath. The second category aims to predict the exact QoT performance in order to compete with analytical models. The third category uses AI to improve the performance of QoT estimation models by reducing the uncertainty on input parameters. The last category consists of improving the performance and generalization ability of AI-based solutions by retraining the models using the least amount of training samples through transfer learning techniques.

QoT models can act as part of the digital twin of the operational network by simulating the impact of new network configurations before deploying them. However, they require a perfect knowledge of the network state, consisting of a set of optical parameters that have different levels of uncertainty. Using the QoT measurements collected by the network controller, we can have a feedback about the QoT estimation inaccuracy, which can potentially be addressed using ML based techniques. In this thesis, we study this issue of uncertainty in network parameters and consider three approaches that can improve the QoT estimation in this case. We propose different learning processes in each approach, and test their performance using simulation and real data.

The first approach relies on optimizing the network parameters using the QoT estimation error as an objective function. We apply this approach through two learning processes to target QoT estimation tools based respectively on analytical model (GNPy) and Machine Learning (neural network). This approach can minimize the SNR estimation error to close to 0 dB on already trained network configuration, and reaches 0.3 dB estimation error on unseen network configurations.

In the second approach, we retrain a neural network based model to adapt it to changes in QoT due to parameters uncertainty through Transfer Learning. We show how the model can relearn the new behavior of the network without searching for the correct values of the network parameters. We can reach up to 0.5 dB in validation error with only ten new training samples.

The last approach consists of detecting parameter changes in response to failure events using reinforcement learning techniques. We consider two types of failure events. We show that the model can correctly classify the events with up to 93% of accuracy in small network topologies.

Finally, we apply the first approach based on Bayesian Optimization algorithm to refine network parameters using data collected from a live network. We use data extracted from the north-bound interface of the network controller to build a network state based on the input data model of GNPy. Then, we apply our learning process on two transmission lines, which led to SNR estimation improvement up to 1.7 dB for the monitored services.

Keywords

Quality of Transmission, Machine Learning, Artificial Intelligence, Parameter Optimization, Optical Networks.

Résumé

L'estimation de la qualité de transmission (QoT) des chemins optiques est cruciale dans la conception du réseau et le provisionnement des services. Des études récentes se sont tournées vers les techniques de l'intelligence artificielle (IA) pour améliorer la précision de l'estimation de la QoT, en utilisant les données générées par le réseau optique. Nous distinguons quatre catégories de solutions. La première catégorie consiste à construire un modèle d'IA pour vérifier la faisabilité d'un chemin optique. La deuxième catégorie vise à proposer des modèles basés sur l'IA pour remplacer les modèles analytiques. La troisième catégorie utilise l'IA pour améliorer les performances des modèles d'estimation de la QoT en réduisant l'incertitude sur les paramètres d'entrée. La dernière catégorie consiste à améliorer les performances et la capacité de généralisation des solutions à base d'IA en améliorant les échantillons des jeux de données dans la phase d'apprentissage grâce à des techniques d'apprentissage par transfert.

Les modèles d'estimation de la QoT peuvent constituer un module dans le Digital Twin du réseau optique, visant à simuler l'impact d'une nouvelle configuration sur la performance du réseau avant la phase de déploiement. Cependant, ces modèles requièrent une connaissance parfaite de l'état du réseau, représenté à partir d'un ensemble de paramètres optiques ayant des valeurs qui peuvent être certaines ou incertaines. Les mesures de performance collectées par le contrôleur peuvent représenter un feedback sur la précision de l'estimation de la QoT, ce qui peut déclencher des algorithmes à base de machine learning pour raffiner les valeurs des paramètres incertains. Dans cette thèse, nous étudions le problème d'incertitude des paramètres, et nous proposons trois approches pour améliorer la QoT dans ce cas. Nous proposons pour chaque approche un certain nombre de processus d'apprentissage et nous testons leur performances avec des données de simulation et des données collectées à partir du réseau opérationnel.

La première approche se base sur l'optimisation des paramètres du réseau en utilisant l'erreur dans l'estimation de la QoT comme fonction d'objectif. Nous implémentons cette approche avec deux processus d'apprentissage, le premier basé sur un modèle analytique (GNPy) et le deuxième sur un modèle à base de machine learning (réseau de neurones). Cette approche arrive à minimiser l'erreur d'estimation jusqu'à 0 dB pour des configurations de réseau où le modèle a été déjà entraîné, et atteint une erreur d'estimation de 0.3 dB sur des nouvelles configurations.

Dans la deuxième approche, nous ré-entraînons un modèle basé sur un réseau de neurones pour l'adapter à l'incertitude des paramètres en utilisant l'apprentissage par transfert. Nous montrons que le modèle peut apprendre un nouveau comportement sans optimiser les paramètres incertains. Nous arrivons à une erreur de validation de 0.5 dB avec seulement dix nouveaux échantillons.

La dernière approche consiste à détecter les changements de paramètres en réponse à un événement de panne en utilisant des techniques d'apprentissage par renforcement. Nous considérons deux types de pannes, et nous montrons que le modèle atteint une précision de

classification de 93% dans une petite topologie.

Enfin, nous appliquons la première approche basée sur l'optimisation Bayésienne pour raffiner les paramètres du réseau sur des données collectées à partir d'un réseau opérationnel. Nous extrayons les données à partir de l'interface nord du contrôleur, et nous construisons l'état du réseau selon le modèle de donnée d'entrée de GNP_y. En appliquant notre processus d'apprentissage sur deux lignes de transmission, nous constatons une réduction d'erreur qui atteint 1.7 dB sur les services monitorés.

Mots-clés

Qualité de Transmission, Machine Learning, Intelligence Artificielle, Optimisation des Paramètres, Réseaux Optiques.

Table of contents

1	Introduction	19
1.1	Context and Motivation	20
1.2	Objectives of the Thesis	22
1.3	Contributions of the Thesis	22
1.4	Publications List	23
1.5	Relationship of Publications with Contributions	24
1.6	Outline of the Thesis	25
2	Literature Overview	27
2.1	Introduction	28
2.2	QoT Estimation Approaches	28
2.2.1	Transmission impairments	28
2.2.2	QoT indicators	29
2.2.3	Physical layer models	30
2.3	ML-based QoT solutions	31
2.3.1	ML based models to check lightpath feasibility	32
2.3.2	ML based models to estimate a lightpath's QoT	34
2.3.3	ML models to enhance analytical models	35
2.3.4	ML techniques to improve model generalization	37
2.4	Discussion	39
2.4.1	Adopted approaches	39
2.4.2	Dataset collection	41
2.4.3	ML models and input/output features	43
2.4.4	Performance and evaluation metrics	44
2.5	Conclusion	45
3	Problem Description	47
3.1	Introduction	48
3.2	Problem Description and Proposed Framework	48
3.3	GNPy as Analytical QoT Estimator	51
3.4	Neural Network as QoT Estimator	52

3.5	Selection of Uncertain Parameters	54
3.5.1	Simulation Description	54
3.5.2	Service-Related Parameters	55
3.5.3	Transponder-Related Parameters	55
3.5.4	ROADM-Related Parameters	55
3.5.5	Amplifier-Related Parameters	56
3.5.6	Fiber-Related Parameters	58
3.5.7	Optical Power	60
3.5.8	QoT metrics	60
3.5.9	Discussion on parameters uncertainty	60
3.6	Conclusion	61
4	AI Techniques Comparison Using Synthetic Data	63
4.1	Introduction	64
4.2	Problem Formulation	64
4.3	Simulation Network Topology	65
4.4	First Approach: Optimizing Network Parameters	66
4.4.1	LP1 - Bayesian Optimization for Analytical QoT Model	67
4.4.1.1	Simulation Scenario 1: Optimization Without Power Measurements	70
4.4.1.2	Simulation Scenario 2: Optimization With Power Measurements	72
4.4.1.3	Simulation Scenario 3: Optimization with Power Measurements and Signal Ripple Compensation	75
4.4.2	LP2 - Gradient Descent for ML-based QoT Model	76
4.4.2.1	Simulation Scenario 1: Comparing SNR Estimation Using NN and GNPY	78
4.4.2.2	Simulation Scenario 2: Optimization With Power Measurements	79
4.4.2.3	Simulation Scenario 3: Optimization With Power Measurements and Signal Ripple Compensation	82
4.5	Second Approach: Adapting the QoT Model	84
4.5.1	LP3 - Transfer Learning to Adapt a ML-Based QoT Model	84
4.5.1.1	Simulation Scenario 1: Retraining the Model Without Knowing Uncertain Parameters	86
4.5.1.2	Simulation Scenario 2: Retraining the Model With Knowledge of Uncertain Parameters	89
4.5.1.3	Discussion Around LP3	91
4.6	Third Approach: Detecting Parameter Changes	92
4.6.1	LP4 - Detecting parameter changes using reinforcement learning	94
4.6.1.1	Simulation Scenario 1: Classification with 3 possible events	95
4.6.2	Discussion Around LP4	97
4.7	Conclusion	98

5	Field Trial to Assess Bayesian Optimisation Based Approach	99
5.1	Introduction	100
5.2	Data Extraction	100
5.2.1	Extraction Tool	100
5.2.2	Data Description	102
5.2.3	Experiment Sub-Network Topology	104
5.3	Experiment Results	105
5.4	Conclusion	108
6	Conclusion and Future Work	109
6.1	Conclusion	110
6.2	Perspectives	114
	References	116
	List of figures	128
	List of tables	131
	Appendix133	

Chapter **1**

Introduction

Contents

1.1	Context and Motivation	20
1.2	Objectives of the Thesis	22
1.3	Contributions of the Thesis	22
1.4	Publications List	23
1.5	Relationship of Publications with Contributions	24
1.6	Outline of the Thesis	25

1.1 Context and Motivation

In a constant attempt to meet increasing capacity demands, optical transport networks have steadily evolved through a number of technological advances. Technologies such as coherent transmission, flexible modulation and tunable transceivers have led to a plethora of new parameters and configurations that complicate network design and operation. At the same time, revolutionary initiatives are emerging with the introduction of Software Defined Networks (SDN) [1] that could open up new opportunities to deal with these complexities in optical networks. Among these initiatives are those that push towards the promotion of the openness of Application Programming Interfaces (APIs) and the definition of common data models (e.g., T-API [2] and OpenROADM initiative [3][4]). Other initiatives propose new control and monitoring protocols such as Netconf and gRPC [5][6]. These solutions will lead to providing large amounts of data in standardized format that could be harnessed to solve network issues using new paradigms in optical networks like Artificial Intelligence (AI).

AI is the introduction of intelligence to machines in order to perform complex tasks in a similar manner to humans. The most popular sub-field of AI is machine learning (ML). ML consists of algorithms that capture patterns and behaviors in the data in order to produce models for a variety of tasks such as estimating a value based on inputs (i.e., regression techniques) and classifying data into groups (i.e., classification techniques). ML has seen an increase in popularity in research in recent years in multiple fields, particularly computer vision, natural language processing and speech recognition [7]. ML is subject to the same scrutiny in optical networks. A large amount of papers have been published on the application of ML techniques to multiple use cases: routing and wavelength (RWA)/spectrum assignment (RSA) [8][9], Quality of Transmission (QoT) estimation [10] and fault management [11].

QoT estimation is of particular interest for optical networks. It consists of assessing the performance of an existing or candidate lightpath based on its characteristics and the network configuration. The QoT is used to monitor the health of an existing lightpath or check the feasibility of a candidate one by comparing its predicted QoT to the receiver's threshold. Estimating the QoT of a lightpath is crucial in network design and service provisioning. In fact, an underestimated QoT value can lead to significant loss in capacity and increase the network deployment cost (e.g., unnecessary equipment expenditure). On the other hand, an overestimation of QoT can lead to unstable lightpath. QoT estimation is also the basis of network optimization, as an accurate QoT is required for optimal RWA/RSA and capacity maximization.

The difficulty of QoT estimation stems mainly from the various impairments in the

fiber that optical transmission is subject to [12]. Linear impairments are due to the signal attenuation, chromatic/polarization dispersion and the noise generated by the equipment. Nonlinear impairments include effects such as Kerr and scattering effects. QoT estimation must also take into consideration the behavior of various transmission equipment that vary widely in their performance according to their models, types and vendors. QoT estimation is generally performed using analytical models. An analytical model features a model of the fiber transmission also called a Physical Layer Model (PLM) that estimates the linear and nonlinear impairments. Analytical models vary in their accuracy, execution time and considered assumptions which always leaves room for improvement. ML is being used to improve QoT estimation in several creative ways such as building ML-based QoT estimators, modeling the response of optical equipment or estimating nonlinear effects.

An accurate estimation of Quality of Transmission (QoT) is a critical component in developing a Digital Twin (DT) [13]. A DT serves as a software replica of the actual network, enabling simulations of changes before implementing them in the real network. The current trend in literature focuses on constructing DTs for Information Technology (IT) systems, which facilitates network automation, optimization, and fault management. To create a DT for QoT estimation, it is necessary to represent the network using parameters (referred to as network state) and employ a precise QoT estimation tool. By considering QoT estimation within the specific context of DT development, we can define the task of QoT improvement more accurately. In fact, the QoT estimation must ensure sufficient accuracy to align the simulated changes in the DT with the actual effects on the network once implemented. Furthermore, the QoT estimation should adapt to any potential inaccuracies in the parameters used to construct the network state of the DT. This DT approach helps bridge the gap between QoT estimation under ideal design conditions and the real network state, empowering network operators to exercise better control over the network.

From an operator's perspective, QoT estimation poses significant challenges. In today's networks, equipment is typically sourced from different suppliers, each with their own parameter specifications, data models, and protocols for accessing network data. These elements are often proprietary, making it difficult for operators to accurately represent the network state. Additionally, QoT estimation in live networks relies on black-box supplier-proprietary tools, further complicating operational tasks. Moreover, the live network experiences variations due to faults and equipment aging, requiring QoT tools to be used beyond their ideal prediction conditions. In this context, ML can enhance QoT estimation by learning network behavior directly from data. This approach has the potential to overcome challenges related to parameter availability, network state accuracy, and the limitations of existing QoT estimation models.

1.2 Objectives of the Thesis

The main objective of this thesis is to explore ways to improve the estimation of QoT in optical transport network using data based algorithms. This objective comes from a realization that optical networks control plane has a wealth of data that can be potentially used to improve operational functions and modules such as the QoT estimation module. This comes at a time where machine learning is being used in a multiple application fields to yield great results in terms of automation, performance gain and cognitive breakthroughs. After a rigorous literature overview, we settled on the topic of improving QoT estimation by reducing uncertainty on network parameters. We then set the following objectives for this thesis:

- Survey recent publications to find ways to improve QoT estimation using ML techniques.
- Propose learning processes to improve QoT estimation using network data.
- Propose simulation scenarios to test the learning processes.
- Validate the learning processes in a real operational scenario using data collected from the a network.

These four goals were addressed in several contributions as outlined in the following section.

1.3 Contributions of the Thesis

In order to meet the above research objectives, we have proposed three main contributions. The first contribution is a rigorous literature overview that shows how ML techniques can be used to improve QoT estimation. In the second contribution, we propose different data-based learning processes to improve QoT estimation. We then validate them using different simulation scenarios. More specifically, we propose in this contribution four learning processes. Two learning processes to optimize network parameters using both ML and analytical based QoT models. One learning process based on transfer learning techniques to improve the QoT estimation of a ML-based model. The last learning process is based on a reinforcement learning technique to detect parameter changes in the network. The last contribution consists of testing the proposed learning processes and validating their relevance using data collected from a live network.

- C.1 The first contribution is a rigorous literature overview that shows how ML techniques can be used to improve QoT estimation.

C.2 In the second contribution, we propose different data-based learning processes to improve QoT estimation and validate them in different simulation scenarios. More specifically, this contribution provides three sub-contributions, as follows:

C.2.1 Firstly, it proposes two learning processes to optimize network parameters using both ML and analytical based QoT models.

C.2.2 Secondly, it proposes a transfer learning technique to improve the estimation of a ML-based QoT estimation model.

C.2.3 Thirdly, it proposes a reinforcement learning technique to detect parameter changes in the network.

C.3 The last contribution consists of the validation of the proposed learning process using data collected from a live network.

1.4 Publications List

Journal Papers

- [14] AYASSI Reda, TRIKI Ahmed, CRESPI Noel, MINERVA Roberto and LAYE Maxime “Survey on the Use of Machine Learning for Quality of Transmission Estimation in Optical Transport Networks”, *J. Lightwave Technol.* 40, 5803-5815 (2022)
- [15] BABAR Jatin, AYASSI Reda, TRIKI Ahmed, and LAYE Maxime, “Machine learning models for alarm classification and failure localization in optical transport networks”, *J. Opt. Commun. Netw.* 14, 621-628 (2022)

Conference Papers

- [16] AYASSI Reda, TRIKI Ahmed, LE GUYADER Bertrand, FRANK Florian, LE ROUZIC Esther, HERVIOU Fabrice, CRESPI Noel and MINERVA Roberto. “Field Trial to Assess Bayesian Optimization for Improving QoT Estimation”. In: European Conference on Optical Communication. *Optica Publishing Group*, 2023.
- [17] AYASSI, Reda, TRIKI, Ahmed, LAYE, Maxime, LE ROUZIC Esther, CRESPI Noel and MINERVA Roberto. “Bayesian optimization-based algorithm to improve the quality of transmission estimation”. In: Photonic Networks and Devices. *Optica Publishing Group*, 2021. p. NeF2B. 3.
- [18] AYASSI Reda, TRIKI Ahmed, LAYE Maxime, CRESPI Noel, MINERVA Roberto and CATANESE Clara. “An overview on machine learning-based solutions to improve

lightpath QoT estimation”. In: 2020 22nd International Conference on Transparent Optical Networks (ICTON). *IEEE*, 2020. p. 1-4.

- [19] CATANESE Clara, AYASSI Reda, PINCEMIN Erwan and JAOUEN Yves. “A Fully Connected Neural Network to Mitigate 200G DP-16-QAM Transmission System Impairments”. In: Signal Processing in Photonic Communications. *Optical Society of America*, 2020. p. SpTh3I. 1.
- [20] CATANESE Clara, AYASSI Reda, PINCEMIN Erwan and JAOUEN Yves. “A fully connected neural network approach to mitigate fiber nonlinear effects in 200G DP-16-QAM transmission system”. In: 2020 22nd International Conference on Transparent Optical Networks (ICTON). *IEEE*, 2020. p. 1-4.

1.5 Relationship of Publications with Contributions

In this section, we provide the relationships of publications with contributions.

- The publications ‘*An overview on machine learning-based solutions to improve light-path QoT estimation*’ and ‘*Survey on the Use of Machine Learning for Quality of Transmission Estimation in Optical Transport Networks*’ corresponds to Contribution C.1 in chapter 2.
- The publication ‘*Bayesian optimization-based algorithm to improve the quality of transmission estimation*’ corresponds to Contribution C.2.1 in Section 4.4.
- The publication ‘*Field Trial to Assess Bayesian Optimization for Improving QoT Estimation*’ corresponds to Contribution C.3 in Chapter 5.
- The two publications ‘*A Fully Connected Neural Network to Mitigate 200G DP-16-QAM Transmission System Impairments*’ and ‘*A fully connected neural network approach to mitigate fiber nonlinear effects in 200G DP-16-QAM transmission system*’ were the results of a collaboration with a colleague on the topic of non-linear effects mitigation using ML algorithms. This collaboration is not directly related to the main topic of this thesis.
- The publication ‘*Machine learning models for alarm classification and failure localization in optical transport networks*’ is the result of a collaboration with an intern on the topic of alarm classification in optical networks. This topic is closely related to the QoT estimation improvement topic, and will be addressed in future work by different colleagues.

1.6 Outline of the Thesis

The thesis is structured into 6 chapters:

- **Chapter 1** is the current chapter and serves to introduce the context and the scope of the thesis.
- **Chapter 2** includes a literature overview of recent papers that use ML techniques to improve QoT estimation.
- **Chapter 3** describes the QoT estimation framework that we chose to study and discusses optical parameters uncertainty.
- **Chapter 4** presents the proposed learning processes and the simulation results related to their performance.
- **Chapter 5** describes an experiment on a live network where data are extracted and used to optimize the QoT using a learning processes based on Bayesian Optimization.
- **Chapter 6** summarizes the thesis and discusses possible perspectives and benefits from the obtained results.

Literature Overview

Contents

2.1	Introduction	28
2.2	QoT Estimation Approaches	28
2.2.1	Transmission impairments	28
2.2.2	QoT indicators	29
2.2.3	Physical layer models	30
2.3	ML-based QoT solutions	31
2.3.1	ML based models to check lightpath feasibility	32
2.3.2	ML based models to estimate a lightpath's QoT	34
2.3.3	ML models to enhance analytical models	35
2.3.4	ML techniques to improve model generalization	37
2.4	Discussion	39
2.4.1	Adopted approaches	39
2.4.2	Dataset collection	41
2.4.3	ML models and input/output features	43
2.4.4	Performance and evaluation metrics	44
2.5	Conclusion	45

2.1 Introduction

Estimating the Quality of Transmission (QoT) of the optical signal from source to destination nodes is the cornerstone of design engineering and service provisioning in optical transport networks. Recent studies have turned to Machine Learning (ML) techniques to improve the accuracy of QoT estimation. In this chapter, we survey the literature on this topic and classify the studies into categories based on their scope. Accordingly, we distinguish four categories of ML-based solutions: i) check lightpath feasibility, ii) estimate a lightpath's QoT, iii) enhance existing analytical models and iv) improve model generalization. We describe the proposed solutions in each category in terms of ML algorithms, inputs/outputs of the models, source of data and performance evaluation. Deploying a ML-based solution in the real field is not straightforward and presents several challenges. Therefore, we also discuss from an operator's perspective the potential of these solutions for real-field deployment.

This chapter is structured as follows. Section 2.2 gives an in-depth look into the QoT estimation problem from an analytical point of view. Section 2.3 gives an overview of papers that apply ML to estimate QoT in terms of ML models, used data and the obtained performances. Section 2.4 discusses the challenges of the proposed solutions and their applicability in the real field. Finally, we conclude the chapter in section 2.5 by presenting our main take away, and how this literature impacted the rest of our research efforts. The contribution presented in this chapter were published in a conference paper [18] and a journal paper [14].

2.2 QoT Estimation Approaches

QoT estimation is traditionally performed using analytical models. An analytical model is built upon four elements: i) the transmission impairments taken into consideration, ii) the QoT indicator to be estimated and iii) the physical layer model (PLM) used to model the transmission and iv) the information about the transmission required as input. In this section, we provide an overview of each of these factors in order to show the current challenges of analytical QoT estimation.

2.2.1 Transmission impairments

Impairments in the transmission come either from the propagation of the signal through the fiber, or from the behavior of optical equipment. Fiber optical impairments can be split into linear and nonlinear effects. Linear effects include signal attenuation, Chromatic Dispersion (CD), Polarization Dependent Loss (PDL) and Polarization Mode Dispersion

(PMD). Signal attenuation is rectified using optical amplifiers, however, this degrades the signal by adding Amplified Spontaneous Emission (ASE) noise. CD, PDL and PMD, on the other hand, are compensated using modern digital signal processing (DSP) techniques in the receiver. Nonlinear impairments are either due to the Kerr effects that include Self-Phase Modulation (SPM), Cross-Phase Modulation (CPM) and Four-Wave Mixing (FWM), or the inelastic scattering phenomenon that includes Stimulated Brillouin-Scattering (SBS) and Stimulated Raman-Scattering (SRS). The scattering effects manifest themselves as a tilt in the spectrum which can be corrected using power equalizers. Kerr effects are usually modeled using equations that calculate the Power Spectral Density (PSD) of the signal such as the Schrödinger or Manakov equations [12]. Significant research effort is dedicated to the mitigation of nonlinear effects, and ML-based solutions have recently been proposed for this purpose [21].

In addition to propagation impairments, equipment generate impairments that contribute to signal degradation. The ASE noise generated by amplifiers, especially Erbium-Doped Fiber Amplifiers (EDFA), significantly degrades the signal [22]. Additionally, wavelength and polarization dependent gain in the amplifiers introduces a tilt and ripple effect on signal spectrum. The impairments in a Reconfigurable Optical Add-Drop Multiplexer (ROADM) include PMD and PDL effects, insertion losses, ASE noise from internal amplifiers, filtering effects from imperfect filters and cross-talk effects between the channels. The impairments from the equipment are estimated by doing laboratory characterization, or by modeling each equipment analytically.

2.2.2 QoT indicators

QoT is generally measured using either the Bit Error Rate (BER) or the Signal to Noise Ratio (SNR). SNR represents the ratio of the power of the optical signal to the noise contribution of all the optical impairments mentioned above. Linear SNR, referred to as Optical SNR (OSNR), is defined as the ratio of optical power of the signal P_{sig} to optical noise added to the signal by optical amplifiers P_{ase} as in 2.1 (see 6.2).

$$OSNR = \frac{P_{sig}}{P_{ASE}} \quad (2.1)$$

The OSNR can be measured using an optical spectrum analyzer (OSA) [23], which is not possible for the SNR. BER, on the other hand, is a measure of the number of errors in the received bits. A lightpath is considered healthy if its BER is above a certain threshold and the receiver's Forward Error Correction (FEC) module is able to correct the error in the bits. SNR is computed before the deployment of a lightpath to check its feasibility taking into account optical impairments as well as various margins such as end of life margins and

equipment aging [24]. BER is measured at the transceiver in real time, so it can be used to monitor the health of a lightpath. Before service deployment, BER cannot be estimated but deduced from the SNR once the modulation format, and the transceiver's back-to-back penalty are provided. This characterization is done by mapping the back-to-back OSNR to the BER response of the transceiver [25].

2.2.3 Physical layer models

Analytical models are based on PLMs that attempt to model the propagation of the signal through the fiber medium. This generally comes down to estimating the PSD that is defined in the Schrödinger equation. Each PLM takes into consideration a number of impairments based on the assumptions taken into consideration. A large number of models have been proposed in literature [26]. We distinguish two families of QoT estimation analytical models. The first family consists of exact models that use comprehensive and extremely accurate methods. These models are heavy to execute and require a large number of parameters to model the transmission line. Therefore, they are more suitable for laboratory simulations as their execution time and parameter requirements make them inconvenient to be used in the field. Among the models of this family, we find the Split-Step Fourier method [27] (SSFM) which is a numerical method of solving Schrödinger's equation by splitting the transmission into a succession of small linear and nonlinear steps. It is highly flexible and can be used to simulate network scenarios that have not yet been deployed. Its high computational requirements make this method unsuited for online QoT estimation.

The second family consists of approximate models that are able to estimate the QoT accurately once a set of assumptions are satisfied. The most popular class of these models are the ones that consider nonlinear interference as a small perturbation of the signal. Among perturbation models, we find models based on truncated Volterra Series [28], logarithmic perturbation models [29] as well as Gaussian Noise (GN) models [30]. The light computational load makes these models more likely to be used in an operational context. The GN-model [30] for instance considers that the nonlinear interference in the fiber can be modeled as white Gaussian noise. It is based on three main assumptions [30]: i) nonlinear noise is a perturbation of the signal, ii) the transmitted signal statistically behaves as stationary Gaussian noise and iii) interference in the fiber is an additive Gaussian noise. These assumptions simplify the expression of the PSD defined by the Manakov equation (which itself is a simplification of the Schrödinger equation [31]). In GN-model, the SNR is redefined as the generalized SNR (GSNR) as in 2.2.

$$GSNR = \frac{P_{sig}}{P_{NL} + P_{ASE}} \quad (2.2)$$

Several versions of the GN-model have been proposed such as the enhanced GN-model (EGN) [32] which removes the third assumption (iii), or the generalized GN-model (GGN) [33] which includes the SRS noise contribution. Analytical models based on the GN-model are fast to compute and have been experimentally demonstrated to have satisfying results [34]. However, their performance drops when the aforementioned assumptions fail (e.g., in highly nonlinear regimes).

2.3 ML-based QoT solutions

In this section, we provide a survey of papers dealing with QoT estimation using machine learning. We distinguish four categories of solutions: A) check lightpath feasibility, B) estimate lightpath's QoT, C) enhance analytical models and D) improve model generalization. These categories were chosen based on the scope of the problem to be solved from an operational context. In the first two categories, the scope is to provide an alternative to analytical models either in the form of a ML lightpath feasibility decision model (category A) or a ML QoT estimation model (category B). The scope in category C is to improve the performance of existing analytical tools instead of replacing them. Solutions in category D aim to make ML based solutions more usable in an operational context by improving model generalization and solving dataset collection issues.

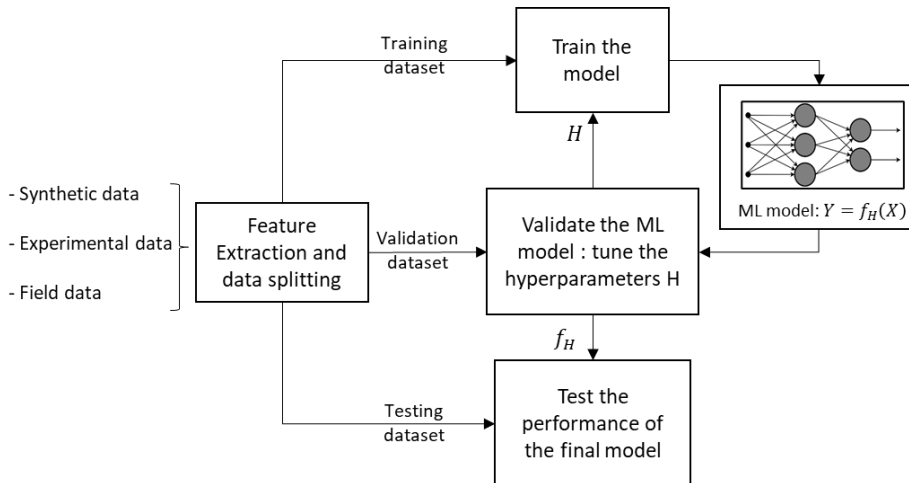


Figure 2.1: General process to train a ML QoT estimation model

The general learning process to train a ML-based QoT estimation model is shown in Figure 2.1. A dataset must be collected using either a simulation, experimental set-up or operational network. In each case, a set of features is extracted from the data and are used as an input of the ML model. The features are selected by cleaning up the dataset

(removing noisy data) and selecting the data that has the most impact on QoT estimation. Then, the dataset is split into three smaller datasets to be used in training, validation and testing. Training the model means constructing a function that predicts the target values in the training dataset as closely as possible. The validation step is used to tune the hyper-parameters of the model to improve its performance. In the testing step, the final performance of the model is calculated. This learning process is used to train different types of ML model such as regression and classification models. In section 2.3.1 we survey papers that use classification models to check lightpath feasibility and in section 2.3.2 papers that use regression models to estimate lightpath QoT. Section 2.3.3 focuses on improving analytical models and features a mix of regression models to estimate impairments and optimization algorithms to reduce the uncertainty on network parameters. 2.3.4 introduces ML techniques to improve ML model generalization such as transfer learning and active learning. Tables 2.1, 2.2, 2.3 and 2.4 give an overview of the particularities of the proposed algorithms in each category based on a set of criteria:

- Algorithm: the ML techniques used by the solutions.
- Input: the selected features to feed the machine learning model.
- Output: the output of the ML model. This is generally the QoT indicator that the study seeks to compute.
- Source of data: information about how data was collected (i.e., synthetic, simulated or experimental or real data).
- Results: the key results from the study. As most of the papers provide results from multiple experiments, we choose to only mention the most relevant results.

2.3.1 ML based models to check lightpath feasibility

The goal of a ML classification model is to attribute a class to each data entry composed of a combination of features. In the case of QoT estimation, ML classification models are used in literature to decide if a candidate lightpath is feasible or not based on a set of optical parameters. The classes in this case are generally binary: the lightpath's QoT indicator is beyond a predefined threshold or not. Thus, ML is used as a simple decision tool for lightpath deployment. Table 2.1 lists the characteristics of surveyed papers that fall into this category. The performance of a classification model is usually represented by the accuracy score; the ratio of correctly classified lightpaths to the total number of lightpaths.

A case-based reasoning (CBR) approach is proposed in [35] to classify lightpaths based on a Q-factor threshold, proving that only a simple classification model is required to achieve

Table 2.1: Comparison of papers proposing a ML model to check lightpath feasibility

Ref	Algorithm	Inputs	Outputs	Data Source	Performance
[35]	Case based reasoning	Channel wavelength, launch power, loss per span, number of spans, active lightpaths, total input power, total power of the adjacent channels	Q-factor	Experimental	Accuracy between 79% and 98.7%
[36]	SVM	Number of ROADMs, of links, of fiber spans; length of fiber span, launch channel power	OSNR	Synthetic	Accuracy > 95%
[37]	ANN	Total length, max link length, central frequency, number of allocated slots, modulation format, number of amplifiers, number of links	BER based on slice requirement	Simulated	Accuracy > 90%, depends on slices and class per experiment
[38]	Logistic regression, Decision trees, SVM, random forest, xgboost	Hop lengths, number of channels, hop losses, number of hops, modulation format, bitrate, aggregation-based feature engineering	2 classes: bad and good configuration	Simulated	Best performance at 0.9 Area Under the Curve (AUC)
[39]	K-Nearest Neighbors (KNN), logistic regression, SVM, ANN	Number of hops, number of spans, total length, average link length, maximum link length, average attenuation, average dispersion, modulation format	SNR above threshold	Simulated	Accuracy > 95%
[40]	KNN, random forest	Lightpath length, longest link length, number of links, traffic volume, modulation format, left/right guardband, left/right traffic volume, left/right modulation format	P_{pos} probability that the BER of the lightpath exceeds a predefined threshold	Synthetic	Accuracy up to 96 % on certain topologies
[41]	SVM, ANN	Total link length, span length, launch power, modulation format, data rate	SNR	Synthetic	Accuracy > 99%
[42]	SVM, logistic regression, Classification and regression trees (CART), random forest	Lightpath length, link lengths, wavelength, statistics on co-propagating light paths	BER	Simulated	Best performance at 99.9% accuracy
[43]	Deep graph convolutional neural network	Channel adjacency matrix, lightpath length, max link length, central frequency, number of slots, core identifier, modulation format, number of amplifiers, number of links, BER of the deployed lightpaths.	BER classification based on threshold	Synthetic	Accuracy rates between 92% and 97%

high accuracy scores up to 98.7%. More conventional ML models like Support Vector Machines (SVM) and random forest, are used in [39] [40] [41] [42] [38] and [36]. These two models usually have the best performance. Authors in [43] use a more complex ML model based on a deep graph Convolutional Neural Network (CNN) to model inter-channel interference in multi-core fibers which can classify lightpaths with up to 97% accuracy.

Most studies use end-to-end line features, such as total lightpath length and number of spans. However, authors in [40] prove that using additional features from neighboring channels improve the classification results. In [38], the statistical representations of the features are calculated and used in the classification in order to reduce the number of features without losing information. Almost all studies use the BER as a QoT indicator and choose the FEC limit as the threshold to separate the two classes (i.e., feasible/unfeasible lightpath). Authors in [40] additionally provide a degree of certainty to the classification, which can be used to choose between multiple feasible lightpaths.

2.3.2 ML based models to estimate a lightpath's QoT

Table 2.2: Comparison of papers proposing a ML model to estimate a lightpath's QoT metric

Ref	Algorithm	Inputs	Outputs	Data Source	Performance
[44]	ANN	Per channel launch power, per amplifier [gains, NFs, gain tilts], per span input power	SNR	Experimental	SNR standard deviation < 0.14 dB
[45]	ANN	Launch power, laser bias, per amplifier [input power, output power]	Q-factor	Experimental	Q RMSE < 0.02 dB
[46]	Random forest	Number of links, total length, max link length, traffic volume, modulation format	GNSR distribution	Synthetic	Best performance: 0.02 RMSE
[47]	Decision tree, random forest, NN	Received signal power, NLI, ASE, channel frequency, total length	GSNR	Synthetic	Best performance: 0.16 dB in average prediction error
[48]	ANN	Per channel [power, frequency], number of spans, analytical model output [ASE, nonlinear noise]	SNR	Simulated	Max error < 0.5 dB
[49] [50]	Gaussian process	Wavelength, measured OSNR, OSNR noise	OSNR	Field trial testbed	MSE < 0.7 dB
[51]	ANN	Channel under test [symbol rate, transmit power, distance to channel, number of neighbor channels], total used bandwidth, number of WDM channels, number of spans, span length parameters, average power level	SNR	Trained on synthetic and applied real data	Max error < 0.5 dB
[52]	ANN	Channel, noise power on each link	OSNR	Experimental	Average error < 0.5 dB
[53]	ANN	Source node, destination node, OSNR of required path	Per channel OSNR	Experimental	RMSE < 0.2 dB
[54]	CNN for feature extraction + ANN for prediction	Per channel [Power, ASE, NLI, number of spans, total length]	GSNR	Synthetic	Maximum error = 0.37 dB
[55]	Gaussian process regression	Input power, number of spans, baud rate, inter-channel spacing	BER, Q-factor	Simulated and experimental	Average error < 0.3 dB
[56]	Random Forest, ANN, KNN	Distance, number of spans, ASE noise, nonlinear interference, power	GSNR	Synthetic	MAE score < 0.007

In this category, the scope of the proposed solutions is to estimate the precise value of a QoT indicator. Therefore, ML-regression models are used. The learning process to train such a model is outlined in Figure 2.1. However, this scope can be more challenging than the previous one since the model output space is continuous. Having the exact QoT value allows to compare two feasible potential lightpaths. Regression models are usually scored using an error operator, such as Root Mean Squared Error (RMSE) or Mean Absolute Error (MAE). MAX and MIN errors are also used, as they allow to set network margins. Table 2.2 lists the surveyed papers that belong to this category.

The features used in regression models are more diverse than those used in classification. Basic line features, such as the number of spans or length of the link, are always used, noticeably in [55] and [46]. Additionally, authors in [44] consider per-channel features, for instance input power and amplifier gain for each channel in the link. The per-channels features are generally flattened into a single vector, except in the case of [54], where a

two dimensional CNN is used to obtain a one-vector representation of all the features. Furthermore, [44] [45] use features from multiple points of the line, such as input power at each amplifier.

In terms of QoT indicator, we notice that noise based QoT indicators (SNR and OSNR) are more frequently used than BER. Authors in [46] propose to estimate the distribution of the QoT indicator rather than just a single value. Authors in [44] compare the estimation of a ML model to that of an analytical model and show that ML models perform better than analytical ones for lightpaths in the edge of the spectrum.

While the majority of the surveyed papers are based on synthetic and experimental data, a couple of papers have also used real data. Noticeably, authors in [51] showcase a model trained on synthetic data then tested on real network data.

2.3.3 ML models to enhance analytical models

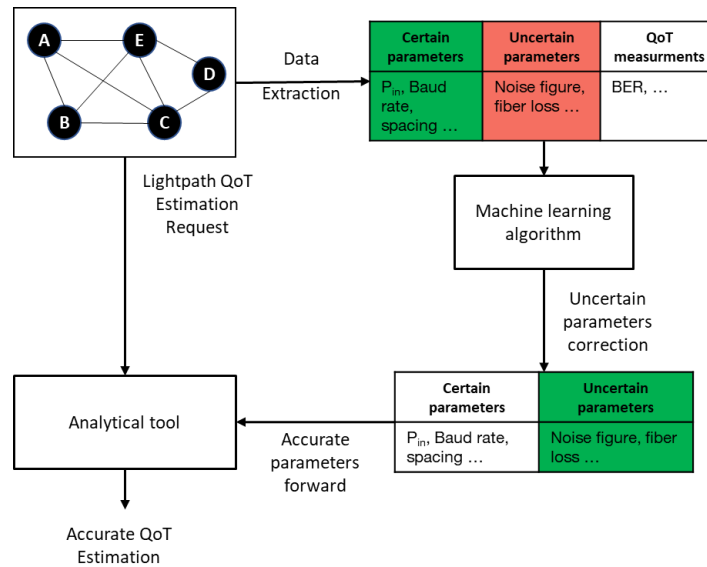


Figure 2.2: Learning process to reduce uncertainty on parameters

The goal of the first and second categories was to build a standalone ML estimator, while the goal of this category is to use ML models and analytical models in tandem. This means that ML is used to improve the accuracy of analytical models instead of replacing them. This can be achieved by, either improving the accuracy of input parameters of an analytical model (i.e., reducing the uncertainty on input parameters) or assessing hard to compute impairments or physical coefficients.

Studies focusing on “Reducing uncertainty on parameters” justify the usefulness of their approach by the fact that some parameters values are not up to date in the operators’

Table 2.3: Comparison of papers attempting to enhance analytical models through machine learning

Ref	Algorithm	Inputs	Outputs	Data Source	Performance
Reducing uncertainty on parameters					
[57]	Gradient descent	SNR	Noise Figure, power profile	Synthetic	Design margins reduced from few dBs to 0.1 dB
[58]	3 step optimizations based on Gradient descent	consecutive power estimation, end to end SNR estimation	Lumped loss before/after fiber spans, amplifier power ripple	Experimental	SNR estimation improved from 2 dB to 0.2 dB
[59]	Custom linear regression	Target Q-factor value, initial noise Figure value, initial nonlinear coefficient value.	Noise Figure, nonlinear coefficient	Experimental	Error reduced from 1.4 dB to 0.6 dB
[60]	Gradient descent	Target SNR value, initial noise Figure value, initial input power value.	Noise Figure, input power	Synthetic	Error reduced by up to 4.18 dB
[61]	Nonlinear curve fitting	Target SNR value, initial values of attenuation, dispersion and non-linear coefficients.	Attenuation, dispersion and non-linear coefficients	Synthetic	Design margin reduction is up to 1.95 dB
[62]	Metropolis algorithm	OSNR estimation	Nonlinear distortion coefficient, filter wavelength detuning, amplifier gain, amplifier noise Figure	Experimental	Error reduction from 3.7 dB to 0.5dB.
Impairment modeling					
[63]	ANN	Spectral load	OSNR	Experimental	Average error < 0.2dB
[64]	ANN	Noise covariance, output of analytical model, number of spans, max span length, average power, launch power, link length, chromatic dispersion, average fiber gamma, average fiber alpha, number of channels.	Nonlinear SNR	Synthetic	0.33 dB of SNR_{nl} deviation using combination of all features
[65]	ANN	Fiber attenuation, dispersion coefficient, effective area, non-linear refractive index	Nonlinear SNR	Simulated	Error below 0.5 dB for 99% of cases
[66]	SVM	Per channel power, optical spectrum, pre-FEC BER	SNR	Synthetic	Average error < 0.2 dB
[67]	Linear Regression, Multivariate polynomial Regression, Decision Tree, Random Forest, Support Vector Machines, K-Nearest Neighbors, NN	Number of fiber spans, span length, channel bandwidth, guard band, number of channels, channel power	SNR (when lightpath length < 200km)	Synthetic	Decreased the cases where the absolute error was higher than 2 dB from 2.30% to 0.47%.

databases because they undergo changes due to multiple factors (e.g., temperature and equipment aging). This could be due to the inability of equipment to measure these parameters (e.g. the fiber nonlinear coefficient) or the inability of the monitoring protocols deployed between the equipment and network management system (NMS) to communicate parameters values in real-time. As an alternative, fixed values like design/beginning-of-life values are used to compute QoT [68]. In literature, optimization algorithms such as gradient descent, are generally used to reduce uncertainty in parameters using the learning process outlined in Figure 2.2. The objective function of the optimization is generally set to the difference between measured and estimated QoT indicator (also called QoT error). Then, the values of a set of uncertain parameters are iteratively changed until the QoT

error is minimal. The amplifier's noise Figure is the most commonly considered uncertain parameter [59]–[62]. The performance of the solution is assessed according to the ability of the algorithm to improve the QoT estimation of the analytical model and reduce the QoT error. This performance closely depends on the number of uncertain parameters assumed in the experiments.

Authors in [59] propose two different approaches for QoT estimation. The first is a purely ML-based estimator that assesses the SNR value. The second approach iteratively reduces the uncertainty of the parameters required as input of an analytical model in order to improve the model's QoT estimation. An elaborate closed-loop controller architecture for optical networks is proposed [61] in order to use feedback from network measurements to improve the accuracy of the ML-model by reducing parameter uncertainty. To improve QoT estimation models, other studies focused on estimating optical parameters that represent hard to compute impairments. This means to assess a certain coefficient required by the analytical model such as the nonlinear coefficient in GN models or by modeling the behavior of an equipment. In Table 2.3, we survey the papers that follow one of these two approaches. Authors in [64] and [65] attempt to estimate the nonlinear SNR. Thus, a neural network model is proposed in [64] to directly estimate the nonlinear noise from line features and fiber characteristics. Authors in [65] propose to mix line features with covariance coefficient calculated from DSP constellations, as well as the output of an analytical model. They show that the nonlinear SNR estimation can be enhanced by feeding all this information as an input of an artificial neural network (ANN). Papers [63] and [66] are focused on modeling the effect of different spectral loads on amplifiers. The aim is to estimate the SNR and OSNR taking into consideration only the impairments generated by the amplifier. This estimation is used alongside an analytical model in order to estimate the overall QoT. In a similar vein, authors in [67] propose a ML model to estimate the SNR of a lightpath. The ML model is used when the total length of the lightpath is inferior to 200 *km*, otherwise, the GNLI is used.

2.3.4 ML techniques to improve model generalization

Model generalization refers to the ability of the model to adapt to data with different distributions. A ML model trained on data extracted from a specific network would not necessarily perform similarly using another network dataset. In order to improve model generalization, more diverse datasets are needed to train the model which is not always possible due to the lack of datasets extracted from different networks. Techniques such as transfer learning and active learning are generally used to resolve this issue. In Table 2.4, we have surveyed papers that try to improve ML model performance using these techniques. It is worth noting that each of these papers propose its own QoT estimation model which

Table 2.4: Comparison of papers featuring machine learning techniques for dataset management

Ref	Algorithm	Inputs	Outputs	Data Source	Performance
[69]	ANN, Active learning through Monte Carlo (MC) dropping for uncertainty sampling	Signal bandwidth, modulation format, peak-to-peak voltage, received power	Generalized mutual information	Experimental	Requires 25% less data than random sampling, while maintaining blow 0.055 in MSE
[70]	DNN, Evolutionary transfer learning	Power profile at each optical switch	Q-Factor	Experimental	Only 10% data size required for retraining
[71]	DNN	DSP constellation	SNR	Experimental	Average Error < 0.2dB
[72]	Gaussian process, Active learning using a MC method, Domain adaptation using Bayesian updating, feature augmentation, Correlation alignment	total lightpath length, longest link length, number of traversed links, traffic volume, and modulation format	SNR	Synthetic	Depends on method and dataset size
[73]	ANN	Span lengths	SNR	Simulated	RMSE improved by 2 after retraining
[74]	DNN	Q-factor of different lines	Q-factor	Experimental	50% less dataset size for retraining
[75]	SVM	Total length, number of links, maximum link length, demand capacity, modulation format	BER classification	Synthetic	20 times less data required for retraining
[76]	DNN	Amplitude histogram of received samples	OSNR	Experimental	20 times less data required for retraining

means that it can also fit into the category A or B. However, we choose to survey them in this category as we consider their main contribution is to solve the model generalization problem. In the characteristics, we provide both the ML algorithm used for QoT estimation and the ML algorithm used for model generalization. The performance is assessed according to the ability to reduce the dataset requirements or to improve error rates. The goal of Transfer Learning (TL) algorithms is domain adaptation, which means ensuring that a ML model is generalizable to multiple datasets with different feature distributions. Figure 2.3 shows how transfer learning can adapt a ML model to multiple network domains. Pesic et al. study in [73] the impact of using networks with different span lengths on the performance of ML-based solution. They show the importance of pre-training a model on unbiased data. This study is extended in [74] to include more network features. Authors also study the effect of domain adaptation on the structure of an ANN model. In [76], [75] and [73], authors use a ML model initially trained on data from a network A in another network B. To adapt the model, a dataset from network B, up to 50 times smaller than that of network A is used for retraining.

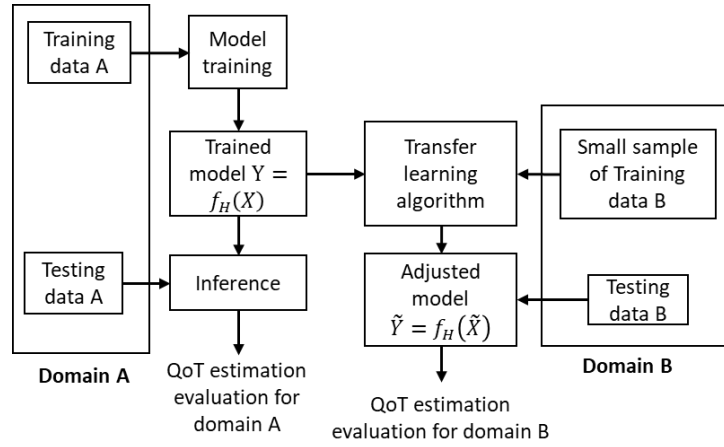


Figure 2.3: Learning process to adapt a ML model from one domain to another using Transfer Learning

Active Learning (AL) approaches are proposed in [69] and [72]. AL algorithms seek to reduce the dataset size by selecting data that best improve model performance. The method typically starts with an initial small dataset, then progressively adds more data entries using an algorithm to compute and rank the importance of each data point. Only the selected data points are measured which avoid the need of a large dataset from the beginning. Both papers use a Monte Carlo based process for data selection. Authors in [72] also propose three domain adaptation techniques, namely Bayesian updating, feature augmentation, correlation alignment, then provide an extensive benchmark of each method.

2.4 Discussion

In this section, we discuss the ML-based solutions described in the previous section. Through this discussion, we present the challenges of these solutions and explore their feasibility in the context of operational networks. We also propose some perspectives for future work. Our discussion focuses on four main points: i) the relevance of the tackled scopes, ii) the dataset used for training, iii) the feature and model choice and iv) the evaluation metrics.

2.4.1 Adopted approaches

The four categories that we have identified to classify the solutions proposed in the literature tackle the use of ML to estimate the QoT in different manners. Each solution has its own benefits and challenges that make it more or less usable in an operational context.

By proposing a ML-based QoT estimator, whether it is a classification or a regression model (Categories A and B), the objective is to provide an alternative to analytical mod-

els. In order to justify the substantial efforts needed to train the model correctly and the extra-costs to deploy such solution, especially when it comes to data collection, ML-based solution must outperform the analytical models in terms of QoT estimation precision, number of required input parameters or/and computational load. Guaranteeing, at least, the performance achieved by analytical model seems to be feasible because the capacity of ML-solutions to model complex physical phenomenon has been extensively proven not only in optics but also in various areas of research [7]. Reducing the execution time is also not a concern, because outside of the training phase, model inference is generally not computationally extensive. In [51], ANN prediction takes microseconds, compared to minutes using a full EGN model. Additionally, ML models may use feature representations with fewer input parameters, for instance by averaging the list of span lengths as in [39]. ML models can also be more robust to uncertain parameters as featured in [77]. However, the challenge facing solutions involved in these categories (i.e., A and B) is to build a model that is effective, easy to maintain and works on all the scenarios where it is applied with acceptable deployment extra-cost.

The choice between classification and regression depends on the operational requirement. If only checking the feasibility of an optical path is needed, classification is better. However, if the value of the QoT is needed (e.g., to compare the performance of different models of equipment), the regression model will be the best option. For both methods, it is useful, from an operational point of view, to additionally assess the estimation error in order to facilitate the computation of operational margins, which might be easier to compute with a regression approach.

The idea behind the category C is to use ML in tandem with analytical models. Analytical models are already used extensively in the field, so proposing solutions to improve their usability is better than replacing them. In fact, missing or inaccurate parameter values is a recurring issue in data extracted from operational networks as outlined in [78]. Studies that focus on reducing the inaccuracy of parameters values have mostly focused on optimization approaches using the error in QoT estimation between measurements and the analytical model as an objective function [57]. The solutions proposed within this scope risk not being able to converge towards the real values because some impairments that are not taken into consideration by the analytical model may contribute to the error in QoT estimation. In this case, alternative values of the input parameters are provided by the ML algorithm which improves the accuracy of the QoT, although the values of input parameters do not correspond to the real values [17].

As analytical models are showing promising results with decent rapidity [68], we believe that solutions that aim to assist analytical models (i.e., Category C) are pragmatic and could provide promising results in short/medium terms. For solutions of categories A and

B, we consider that the main benefit of using a ML estimator rather than an analytical one could be in one of these cases: i) the ML estimator can achieve accurate performance when the analytical model is not appropriate because its assumptions are not satisfied [67], ii) some input parameters of the analytical model could not be provided, whereas they are not needed by ML-based solution (e.g. a better feature representation in [59] leads to reduce the number of input parameters), or iii) the analytical model is unusable due to execution time constraints.

The interest of improving the generalization of ML models (Category D) is justified by the issues around data collection in optical networks. Models proposed within this category do not focus only on computing the QoT itself but to make models proposed under the other categories technically feasible in terms of data and able to be generalized in many scenarios. Concepts presented in this category could be applied to any ML-model that requires training. We believe that this approach will be the key to making ML models a viable solution in the field, especially when it comes to domain adaptation between heterogeneous networks as in [72]. Another point of interest that was rarely addressed in papers is continuous learning. Operational networks are susceptible to change, which might make ML models inefficient if they are not updated regularly. Therefore, proposing a closed-loop process, based for example on active learning as in [79] would help to solve this problem.

2.4.2 Dataset collection

Training ML models requires large datasets. Some approaches do not have a proper training phase, as in the case of solutions based on optimization algorithms such as [59] and [60]. However, even in this case QoT indicator measurements are still required. In general, the performance of ML methods is tightly linked to the quality of the dataset used in the training phase. The different steps to build the dataset which are data collection, data annotation, feature engineering, data augmentation and splitting datasets for training and validation, must be carefully performed in order to ensure a successful training and avoid biased results.

Data collection remains the first bottleneck in optical networks due to several factors: equipment lock-in (i.e., inability to access equipment data), lack of standardized data models and monitoring protocols, lack of data collection and monitoring tools and the cost of deploying optical signal probes in the network. Datasets used in the surveyed papers are either synthetic, experimental or operational. Synthetic datasets are generated by simulating network scenarios using an analytical model, for instance, authors in [46] and [56] use the GNPY tool [80]. This method allows greater control over data entry points, feature variation, flexibility in the definition of the scenario and setting the dataset size. However, models trained with synthetic data learn the behavior of the simulation platform and the

analytical model behind which might not faithfully represent a real optical network behavior. In fact, the performance of the ML in this case is tightly related to the accuracy of the simulation platform. Transfer learning approaches proposed in Section 2.3.4 could solve this problem by retraining the model using field data in order to increase its accuracy and remove the synthetic data bias, but, to our knowledge, this has not yet been demonstrated in the literature.

Experimental setups on the other hand better reproduce the conditions of an operational network, while keeping the flexibility of simulation approaches. For example, in [63], a setup consisting of cascaded amplifiers is used to model amplifier response to spectral load, while in [44] an experimental setup of a full transmission line is used. Through experimental data, models can learn the behavior of a real transmission using physical fibers and equipment. Generating this kind of data is costly and time-consuming given the large number of measurements required to train a ML model. Therefore, a full automation of the experimental setup, as proposed in [81], is highly recommended. The experimental setup is usually limited to a small-scale network. Thus, applying the ML model in a large-scale operational network requires the use of adaption and generalization techniques.

Training ML algorithms with operational data confronts the model to the real condition of the field. However, the data collection process in this case is complex due to the lack of monitoring and data extraction tools in the optical layer and the inability to define on-demand data extraction scenario. For instance, data cannot be extracted from unfeasible or low QoT lightpath and equipment configuration settings cannot be changed for training purposes. Moreover, operational datasets are less diverse in terms of features availability and variation. In fact, these datasets are tightly linked to the network from which they were collected. Since feature distributions could change from one network to another, transfer learning could be applied to generalize the model between heterogeneous networks (or domains) as shown in [74].

We notice that most papers use synthetic or experimental data to train ML models. Only a few studies have used data from an operational network such as [51], [49] and [50] use data extracted from a field trial testbed with a total 436.4 *km* optical path over the national dark fiber facility in UK. The choice of a data source requires a balance between simulation flexibility and representation. The ideal scenario is to have enough variation in field data to train the models correctly. But since this is far from being immediately achievable, we consider that it is more convenient to train models with synthetic or experimental data mixed with samples of operational data to generalize the model's performance.

Data collection concerns hamper research focused on the application of ML in optical networks. While it is justified to adopt an optimist outlook and assume that the data scarcity will be resolved in the future, we believe that it will be more beneficial to actively

tackle the problem by proposing detailed data collection schemes alongside the ML solutions as in [49], or by only taking into consideration parameters that are available in the field as features as in [17].

2.4.3 ML models and input/output features

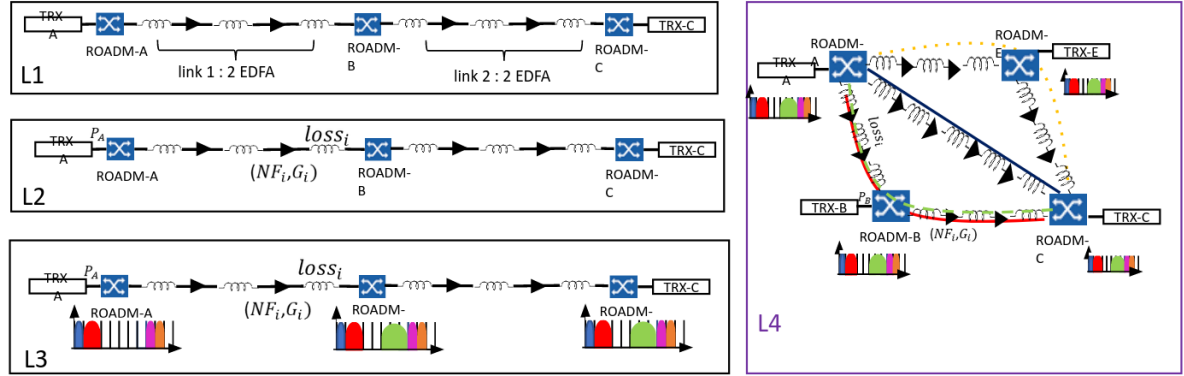


Figure 2.4: Four levels of data features for ML-based solutions

The choice of input features and ML model is motivated primarily by the scope of the solution. If we assume that the goal is to estimate a QoT indicator, the input feature of the model must fully describe the factors that impact this indicator. Similarly, the ML model must be sufficiently complex to model the impact of said features on the indicator. In the surveyed papers, we find different levels of features. The first level concerns the end-to-end lightpath features, such as total length of the lightpath, and number of hops/spans [36]. These features are often related to the lightpath under study rather than considering the entire network. The next level includes data related to specific equipment/fiber through the path such as the attenuation coefficient of the fiber spans, or the amplifier gains in [44]. The third level concerns information about the co-propagating lightpaths (i.e., neighboring wavelengths). These features can range from a simple number of wavelengths to a detailed description of the spectral load [63]. The fourth level is to use a feature representation of the whole network and its lightpaths. This can be modeled using matrices or graphs as in [43]. Feature representation of the network can also be provided with other information depending on the use case: information about slices [37], calculated features from DSP constellations [51], or analytical model output [65]. Setting the level of details in features depends on the impairments to be considered and the level of precision to be achieved by the ML-model. For instance, study [43] proposes a graph based feature representation of all the lightpaths in the networks (i.e., fourth level of feature representation) because they aim to take into consideration inter-core cross-talk effects between all the lightpaths. Figure

2.4 shows the various levels of feature extraction that can be used as input of ML models. Table 2.5 further shows the data that must be extracted from the network in order to extract these features, as well as the operational requirements to collect them. The use of higher level of data probably improves the precision of estimated QoT but it could lead to more complex ML model as well as increase the cost of deploying such solution for a network service provider. For instance, the existing data in the NMS are sufficient to retrieve level L1-data. In this case, the cost needed to deploy such solution mainly concerns the data storage and ML processing. For illustrative purposes, we estimate the rental cost for data storage and execution of algorithm at 10 k euros per month for a ML-solution deployed in one thousand-node network. For levels L2, L3 and L4, streaming telemetry protocol and dedicated hardware are needed to obtain measurement from ports. We estimate the cost of the cards in the order of 100 euros per port.

Table 2.5: Four levels of data features for ML-based solutions

Level	Description	Example	Data extraction requirements
L1	Path routing and lightpath configuration data	Number of EDFA, number of links, average link length, input power ...	Data from the NMS
L2	L1 data + Equipment parameters related to the transmission line	EDFA gains, fiber loss ...	Data from the NMS + streaming telemetry (optional)
L3	L2 data + Routing and spectrum occupancy data related to the line	Neighbor channel bandwidth, per channel power ...	Data from the NMS+ streaming telemetry
L4	L3 for all the network lightpaths of the network	Adjacency matrix	Routing and spectrum occupancy data related to all the network from the NMS+ streaming telemetry

The output of the ML model mainly depends on the scope of the solution. If the output is a QoT indicator that could not be measured in the real network, the only source of data to train and validate the model will be experiments and simulations. Furthermore, the model could not be adjusted in a closed loop architecture during the life of the network. Therefore, the QoT indicator should be carefully chosen to ensure a practical validation of the solution.

2.4.4 Performance and evaluation metrics

Several metrics exist to assess the performance of ML models. The choice of metric depends primarily on the type of model (regression or classification), and on the performance to evaluate. In the case of classification, some studies such as [37] use the accuracy metric, which gives the rate of successful classifications but does not give information on false positives or false negatives. In the case of regression, a Root Mean Squared (RMSE) or/and Mean Absolute Error (MAE) operator are used [46][56]. Using a varied list of evaluation metrics assesses better the model's performance, for instance, providing AUC scores for classification as in [38], and error distribution for regression as in [46]. These evaluation metrics allow to precisely know the model capability which could be helpful for some operational

settings like the specification of network margins.

Furthermore, careful attention must be given to data biases, such as a dataset with higher percentage of a class over another. Multiple papers have used cross validation to detect over-fitting problems like in [56] and [40]. But the best approach is to validate the performance on completely different dataset such as in [51].

In addition to the accuracy of QoT estimation, other evaluation metrics such as network capacity gain [61] or potential resource saving [40] are relevant to show the added-value of the proposed solution. Nevertheless, the assessment of resource saving should also take into consideration the extra cost of deploying the ML solution such as the probes for data monitoring. Finally, a comparison must be established with existing solutions, especially with analytical models in the case of ML-based QoT estimators as in [67].

2.5 Conclusion

Using machine learning to improve the QoT estimation has seen a surge in popularity over the last years. Therefore, we provided in this chapter a survey of studies that investigate this research topic from different angles. We distinguish four categories of models using ML for QoT estimation. The first category consists in building ML model to check the feasibility of a path. The second category aims to make the ML-based model as full alternative of analytical models. The third category uses ML to improve analytical models by either reducing the uncertainty in input parameters and equipment modeling or assessing hard to compute impairments and coefficients to supplement the analytical models. The last category consists in improving the performance and generalization ability of ML-based solution by enhancing the samples of the dataset in the training phase through transfer learning or active learning techniques.

Our takeaway from this literature overview is that QoT estimation models that are totally based on ML models are not sufficiently mature to replace analytical models. The results of the proposed algorithms are generally satisfactory, but some concerns remain regarding their ability to be generalized in order to support complex optical transport network topologies and various equipment configurations. Moreover, the data scarcity and additional cost related to monitoring data and implementing these solutions are among the challenges that hinder the deployment of ML-based QoT estimator in the operational networks.

Therefore, we believe it will be more convenient to focus on reducing optical parameters uncertainty instead of replacing the QoT analytical model by a ML-based one. Indeed, we cannot correctly estimate the QoT if we have a poor knowledge of the state of the network and its parameters. Even, in the case where a ML-based model is used to estimate QoT,

accurate values of input parameters are required. In the remaining sections of this report, we focus on exploring learning process to create a dataset reflecting the actual state of the network using data collected from the NMS. If such a state can be inferred from the measurable and collectable parameters, the estimated QoT will have a high level of accuracy and reliability. Additionally, we explore how QoT models, particularly ML-based ones, can be improved to be able to adapt to changes in network states.

Problem Description

Contents

3.1	Introduction	48
3.2	Problem Description and Proposed Framework	48
3.3	GNPy as Analytical QoT Estimator	51
3.4	Neural Network as QoT Estimator	52
3.5	Selection of Uncertain Parameters	54
3.5.1	Simulation Description	54
3.5.2	Service-Related Parameters	55
3.5.3	Transponder-Related Parameters	55
3.5.4	ROADM-Related Parameters	55
3.5.5	Amplifier-Related Parameters	56
3.5.6	Fiber-Related Parameters	58
3.5.7	Optical Power	60
3.5.8	QoT metrics	60
3.5.9	Discussion on parameters uncertainty	60
3.6	Conclusion	61

3.1 Introduction

QoT estimation tools require the knowledge of a large set of parameters. These parameters can be acquired in different ways and from different sources (e.g., NMS, telemetry, supplier data sheet). Depending on the reliability of the source, some parameters may represent uncertainties in the precision of their values, which leads to inaccuracies in QoT estimation. In section 2.3.3, we showed that AI techniques can be used to refine parameters and provide accurate values of input parameters to analytical tools which improves the QoT estimation. In this chapter, we seek to extend the study of this use case. We first describe the problem in more depth, then we propose a framework where learning processes can be executed to improve QoT estimation in the presence of uncertain parameters. Finally, we study the reliability of different parameters from an operator point of view and assess, through simulation, their impact on the estimation of the QoT.

3.2 Problem Description and Proposed Framework

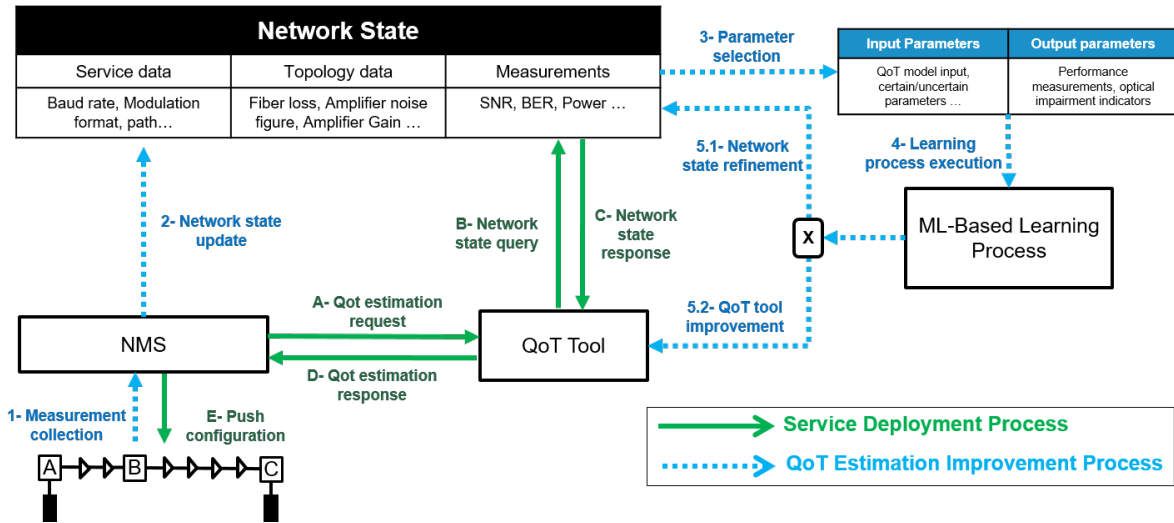


Figure 3.1: QoT computation and improvement framework

From an operator's point of view, QoT estimation is needed to perform the network design and service provisioning. During the network design, the operator must propose a network configuration that guarantees a suitable QoT, in other words a QoT superior to that required by transponders, for all services to be deployed in the network. Network design can involve the establishment of new transmission lines (greenfield scenario) or re-utilize existing parts of the network (brownfield scenario). In the greenfield scenario, we can generally assume that data related to parameters are reliable since a lot of measurements

and tests are performed before the deployment of new transmission lines. However, we can still have some uncertainties related for instance to the inherent imperfection of equipment and connectors. In the brownfield scenario, parameters uncertainties are more prevalent, as parameters related to already-deployed equipment may change during the life cycle of the network. These changes can be unreported by the network controller which impacts the QoT estimation.

In the case of service provisioning, the QoT of each candidate lightpath must be assessed accurately in order to properly route the service. In an operator's context, the transmission lines are already designed to fit a predefined service deployment configuration that estimates how optical services will be provisioned until the end-of-life of the network. However, significant margins are taken into account to compensate uncertainties in parameter and equipment aging. This does not mean that at this stage the service configuration is definitively settled. On the contrary, during the life cycle of the network the provisioning configuration may change to meet the emerging operational needs and new technological advances. For instance, the appearance of flex-grid technology has pushed towards the consideration of new parameters (e.g., number of slots) to compute the QoT. Moreover, obtaining an accurate QoT estimation before service provisioning avoids assuming large system margins which permits network re-optimization and more efficient routing.

In the context of building a DT of the network, the QoT estimation tool can be used to check the performance of an optical service before pushing any changes to the network, and keep an up-to-date vision of the network state. This requires setting up a feedback loop between the controller and the QoT tool to ensure that the QoT indicators that have just been calculated correspond to the measurement (e.g., the BER). Most network operators establish an interface between the QoT estimation tool and the database that could be fed by the controller itself or a dedicated application. However, these data generally suffer from a lack of accuracy (i.e., some data are not correct), reliability (i.e., out-of-date values) and completeness (i.e., missing data). This means that the collected data are rarely used to improve QoT estimation.

In order to improve QoT estimation, we propose a framework as shown in Figure 3.1. The database, hereinafter referred to as *Network State*, holds all the network parameters that are required to calculate the QoT. The network state contains a mix of information that are configured by the operator (e.g. service deployment parameters), measured by the equipment (e.g. power values) or obtained from equipment manufacturing sheets (e.g. amplifier gain spectrum curves). We organize these data into service, topology and measurement data. This network state database is permanently populated with data from the NMS.

The "Service Deployment Process" (represented by green bold arrows in the Figure)

shows the typical scenario where a QoT tool is solicited to estimate the QoT of a potential lightpath. Following a request from the NMS (step A), the QoT tool consults the network state database (step B) in order to get the values of parameters needed to perform the computation of the QoT (step C). Then, the QoT estimation value is sent back to the controller (step D) which uses it to decide which lightpath configuration to deploy (step E). The flow of information from the QoT tool to the controller represents the *feed-forward* part of the QoT estimation closed-loop. Once the deployment is done, new measurements are available in the network such as the BER of the just-deployed lightpath, which can be collected and used to update the network state. Typically, this new information is only stored but not exploited. To close the QoT estimation loop, a second data flow must be set up in order to use the measurements collected in the network to improve the QoT estimation accuracy. Therefore, we propose the "QoT Estimation Improvement Process" which corresponds to the *feedback* part of the closed-loop (represented by blue dashed arrows in the Figure).

The "QoT Estimation Improvement Process" must regularly be performed to adapt to changes in the network measurements. In this process, the network measurements are first collected (step 1), then stored in the network state (step 2). The network state is used to prepare a dataset (step 3). The dataset contains carefully selected data that are required to run the ML-based learning process. In the case of training a ML model for instance, the input could be network features and the output could be QoT metrics. Once the dataset is ready, the learning process is executed (step 4). The learning process typically runs for a few iterations until the maximum improvement is achieved. The outcome of the learning process depends on the chosen approach to solve the problem. In the surveyed papers in chapter 2.1, we identified two typical outcomes (step 5.1 and 5.2). Firstly, the network state can be modified to take into account the changes in network behaviour as proposed in papers reviewed in section 2.3.3 where optical parameters are re-optimized. Secondly, the QoT model can be extended or improved, as proposed in papers reviewed in section 2.3.4 where ML models are retrained to adapt to the characteristics of a new network domain.

As could be noticed, the proposed learning process supposes the existence of measurements, which means that some services have been already deployed. Therefore this solution is applicable to perform network design in the brownfield scenario and in the case of service provisioning. In the case of greenfield network design, the equipment are usually new and well characterized which means they have less parameter uncertainties. In this framework, the AI-based learning process permits a closed-loop control that changes either the network state or the QoT tool so that the estimations match the measurements as much as possible. Whenever a new service is deployed, the available data to run the learning process increase, and consequently the inaccuracy of the QoT estimation is reduced further.

The choice of QoT tool impacts the accuracy of the estimation, the execution time, and the type of parameters that must be collected. In this study, we seek to provide a solution agnostic to the QoT computation tool. Analytical models are the most widely used in the operational network. But this does not preclude that the learning processes should also be applied in the case of neural network-based QoT estimator. With the scale that AI is taking, this kind of solutions are more and more likely to exist. Therefore, we decided to also explore the case where a ML base model is used in the operational network. The final takeaway from our literature review in chapter 2 is that ML models are not mature enough to replace analytical tools in the field today, but this could change in the future due to the accelerated progress of this discipline. The key selling point of ML models is that they can learn the real behaviour of the network from historic data. As a result, the success of these solutions depends on their ability to achieve more accuracy with less computation time than the analytical tool. For this reason, in this chapter we study both cases. For the first case, we take the GNPpy tool ([80]) as a reference analytical tool for QoT estimation. For the second case, we propose and train a neural network (NN) to estimate the QoT. We chose to use GNPpy as it is an open source tool and shows a high efficiency and satisfying results in computing QoT [82]. The comparison between different analytical QoT tools is out of the scope of this study as it does not affect the learning process that we propose.

In the rest of this chapter, we present in section 3.3 and 3.4 the two QoT tools we used in the study, i.e., GNPpy and the neural network model respectively. Then, in section 3.5, we discuss the choice of parameters that constitute the network state and their degree of certainty in an operational context.

3.3 GNPpy as Analytical QoT Estimator

GNPpy is an open source QoT estimation tool based on the Gaussian Noise model. It is developed within the Telecom Infra Project (TIP) by the Physical Simulation Environment (PSE) working group. It is part of the effort to provide a vendor agnostic performance estimator and network planing tool. GNPpy provides two levels of control over QoT estimation; the GNPpy simulation tool, and the GNPpy library. The simulation tool is a high level script that takes as input three types of configuration files (service file, topology file and equipment configuration file) and automatically provides QoT estimation and network design for the desired network. The GNPpy simulation tool is intended for operational users as it requires low configuration effort to run, whereas the GNPpy library is destined to developers as it provides a modular implementation of each element of the optical transmission and offers a higher degree of control over the network simulation.

In our study, we first implemented a new layer over the core GNPpy library called *"LineS-*

imulator". This allows us to use GNPY to simulate multiple network transmission scenarios. Then, for each learning process, we developed a tool that interfaces with *"LineSimulator"*.

GNPY was chosen as the analytical QoT model in our study for multiple reasons. First, it uses the Guassian Noise (GN) model as Physical Layer Models (PLM), and the GN model seems to be the most widely used transmission model in recent literature. Second, it is developed in part by members of our work team which helped in understanding and exploring the tool. Finally, GNPY is an open source tool that aims to be vendor agnostic, meaning that it can be used in a various network scenarios regardless of operator and vendor constrains.

The learning processes we proposed are supposed to be QoT tool agnostic in terms of how QoT is computed but, of course, it must take into consideration the input parameters needed for estimation and the QoT metrics provided. In Appendix 6.2, we present more details of the implementation of GNPY which help in justifying our choice of uncertain parameters in Section 3.5.

3.4 Neural Network as QoT Estimator

In chapter 2, we surveyed several papers that attempt to train ML models to estimate QoT of lightpath (e.g., [44], [45] and [46]). Our conclusion is that in most cases ML models can be successfully trained to achieve a high level of accuracy under specific network transmission scenarios and using specific dataset for training and validation. As far as we know, no ML model can be claimed to be general enough to be applied to estimate the QoT in any use case. However, we can expect more efficient ML models in the coming years due to the availability of data and the high interest of researchers on this topic. For these reasons, we do not seek to provide a highly accurate ML model, nor to outperform existing analytical models. Our goal is to show how parameter uncertainty can affect ML based QoT models, and to provide learning processes to improve QoT estimation in this case.

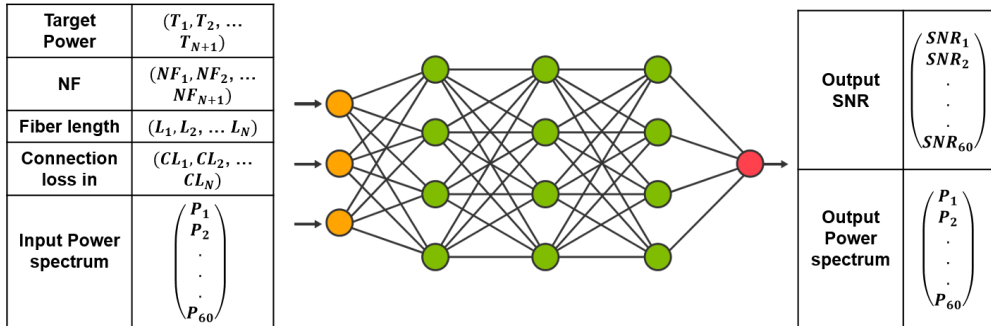


Figure 3.2: Structure of the Neural Network Based QoT Model

Table 3.1: List of Parameters Used to Generate the Training and Validation Datasets

	Parameter Range	Step	Total number of values
Input power spectrum (dBm)	[-2, 4]	0.5	13
Noise Figure (dB)	[4,9]	0.5	11
Connection loss in (dB)	[0,3]	0.2	15
Target Pch out (dBm)	[-1,6]	1	8

We trained a neural network model to estimate the SNR and the power spectrum at the output of an OMS as shown in Figure 3.2. The model is an artificial neural network (ANN) with three hidden layers. We use ReLu activation functions, ADAM [83] optimizer and a weighted Root Mean Squared Error (RMSE) loss function to train the model.

The model takes as input the power spectrum at the input of the booster, the connection in losses of each fiber, the mean NF of each amplifier and the mean Gain of each amplifier. It outputs the SNR spectrum at the end of the OMS, and the power spectrum at the output of the pre-amplifier. In the Section 3.5, we discuss in detail the choice of parameters that impact the QoT estimation. Since our goal is not to train a ML to compete with analytical models, we chose the most relevant input parameters that impact the optimization learning process, as will be explained in the rest of this manuscript.

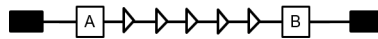


Figure 3.3: Network Topology Used to Generate the Training Dataset for the NN Model

In order to train the model, we use a dataset generated by GNPY. We use a single OMS topology as shown in Figure 3.3 to simulate multiple transmission scenarios by changing the input parameters. In Table 3.1, we show the list of parameters that we change to generate the dataset, as well as the parameters ranges. For each parameter, we take its value range, then split it into 90% training and 10% validation. Then we generate the training and validation dataset by randomly sampling input parameters from the respective training and validation ranges to run the simulation. The output is then calculated with GNPY and added to the dataset. This techniques means that the values of the parameters in the validation dataset will be completely new and have not been encountered during training.

Figure 3.4 shows the summary of the training and validation of the neural network. We do not explore regularization, over-fitting and validation in detail in this study as it is not our main purpose. Figure shows that after 2500 iterations, the RMSE error is below 0.1. This model will be used in the framework presented in Figure 3.1 as a ML-based QoT tool.

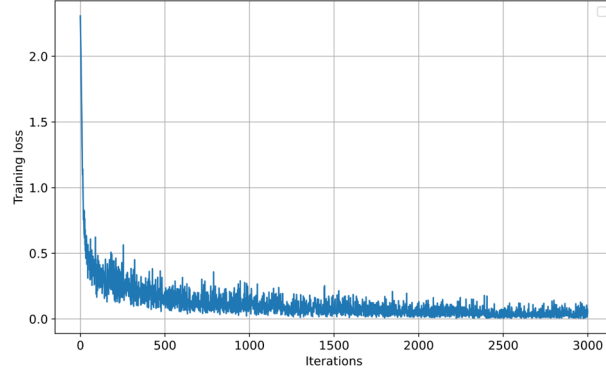


Figure 3.4: Evolution of the training loss during model training

3.5 Selection of Uncertain Parameters

In this section, we discuss the different input parameters of QoT estimation tool in order to distinguish between certain and uncertain parameters and determine which one have the most impact on QoT computation. As the discussion about parameters is strongly related to the QoT tool, we will focus in the section on GNPpy input parameters.

GNPy takes as input a set of parameters in order to compute the above mentioned transmission effect. The core library of GNPpy is split into multiple classes that each represents an equipment line. The transmission is simulated by propagating a spectrum over each element in the line. Since GNPpy is an open-source tool that is still under development, these parameters are subject to change. However, the approach we propose in the following section is designed to be agnostic to the QoT models as discussed before, so it can easily be adapted. We will overview optical parameters in respect to their measurement availability and accuracy according to various hypotheses and observations as a Telecom operator. The information in this section is highly subject to change as measurement technologies progress and is also highly variable based on vendor and operator capabilities.

3.5.1 Simulation Description

To further justify our choice of parameters, we use GNPpy to estimate the impact of each parameter uncertainty on SNR. We use a flat line topology as shown in Figure 3.5 with a full grid of 80 channels over the C-band having a channel spacing of 100 GHz. The span configuration in each OMS is the same: 80 km spans length of Single Mode Fiber (SMF), 16 dB of amplifier gain, and 2 dB of tilt. The per channel output power of each amplifier is set to 4 dBm. We change the values of each parameter that we consider uncertain and

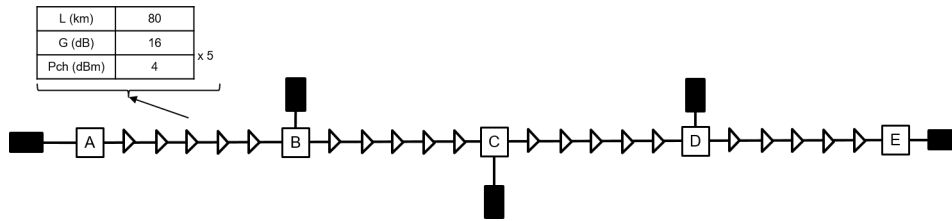


Figure 3.5: Network Topology Used to Calculate Parameter Impact

estimate the SNR using GNPpy. The values of uncertainty used in this chapter are arbitrary and are only there to showcase the impact.

3.5.2 Service-Related Parameters

Those parameters are related to each deployed service including service route, baud rate, modulation format, frequency and bit rate. We generally assume that all of these parameters are available in the network state database with high accuracy. We find in the literature some works such as [84] and [85] that assume the modulation format of an optical service as unknown in the context of alien wavelength but this case is outside the scope of our study.

3.5.3 Transponder-Related Parameters

In GNPpy, almost all transponder-related parameters are taken into account in the service-related parameters. The other transponder-related parameters which remains to be mentioned are the back-to-back noise, the emission power and the central frequency.

The back-to-back noise is modeled in GNPpy as an SNR penalty that is added to the service performance. Actually, this penalty varies between transponders and could be identified using a back-to-back characterization. In our study, we assume that the the back-to-back noise of the transponder is constant and it is set to 45 dB. The output power and the central frequency are configuration parameters and should be available in the controller.

It is worth mentioning that some analytical models calculate the BER instead of the SNR. This requires additional information about the modulation format and constellation shaping. Since GNPpy only estimates the SNR and OSNR, these parameters will not be considered.

3.5.4 ROADM-Related Parameters

ROADM is an optical element capable of switching optical signal in different directions. It is composed of multiple Wavelength Selective Switching (WSS), amplifiers and Variable Optical attenuators (VOA). In order to model the propagation through the ROADM accurately, the parameters related to each components should be taken into consideration (e.g.,

the filtering effects of WSS, noise factor for amplifiers and power attenuation for the VOA). In GNP_y, ROADMs are implemented as an equalization element, with the target output power as a parameter. The target power can be a Power Spectral Density (PSD) value, a flat power value over all channels or a customized power target per channel. The power spectrum at the output of the WSS can potentially be measured if Optical Power Monitors (OPM) are available. Otherwise, we assume that we have knowledge of the equalization method used, such as the PSD value, or the attenuation values for each channel of the WSS. We can use this information to construct the power spectrum at the output of the WSS.

3.5.5 Amplifier-Related Parameters

Three parameters model the EDFA amplifier in GNP_y: the tilt, the gain and the noise figure. Both the noise figure and the gain can be considered as a single mean value, or as a set of parameters, one for each channel. The tilt is configured in the equipment after deployment and generally does not change.

The nominal gain is a configurable parameter in amplifiers. However, the effective gain spectrum can have a number of uncertainties due to gain ripple effects. The effective total gain can be measured if the total input and output powers of the amplifier are available. In this study, we do not consider the gain spectrum as an uncertain parameter, however we do study the network scenario where gain spectrum has a small uncertainty and propose a solution to deal with this issue.

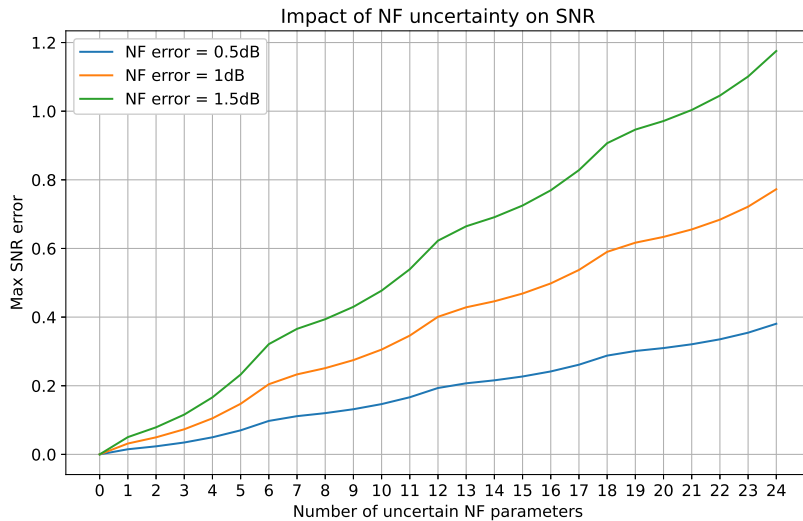


Figure 3.6: Maximum SNR error generated by gradually changing the NF of each amplifier in the 4 OMS FLAT_{_}NET topology

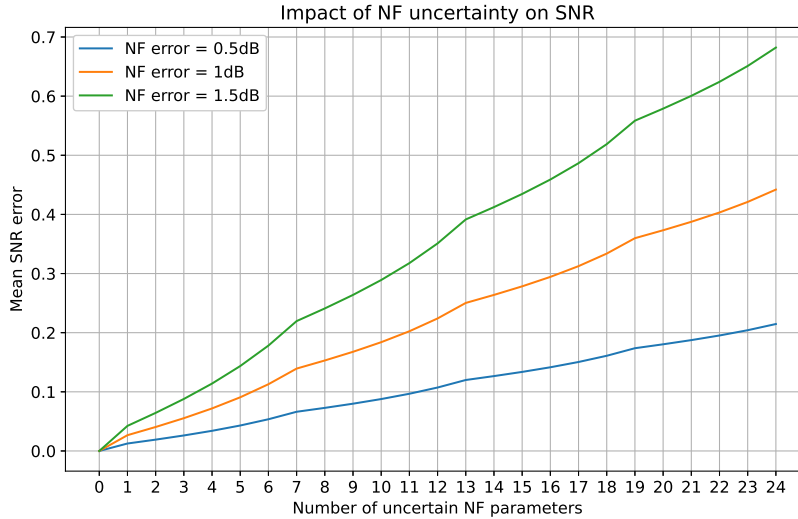


Figure 3.7: Mean SNR error generated by gradually changing the NF of each amplifier in the 4 OMS FLAT_NET topology

Noise Figure (NF) measures the amount of noise added to a signal when it is amplified. A higher noise Figure indicates a poor quality of the amplified signal, and vice versa. We consider the NF as an uncertain parameter. The first reason is that in the data extracted from the operator NMS, the NF values are usually part of the missing data. One solution to obtain these values is to use data sheets provided by some equipment suppliers which contain the NF values corresponding to the gain and input power values. These values, if any, can be inaccurate. The second reason is that the NF represents the main source of ASE noise in the transmission. Including the NF as an uncertain parameter means that it will compensate for uncertainties on ASE noise of the transmission line even if it is not generated by the amplifiers.

In Figures 3.6 and 3.7, we see the evolution of the maximum and mean SNR error when we increase the number of erroneous NF values of the amplifiers that make up the transmission line. We consider the same error on NF for all the amplifiers in each simulation. Three values of error are simulated: 0.5 dB, 1 dB and 1.5 dB. This range is arbitrary as we do not have access to any statistics on NF errors in the field. We notice that at maximum error value, the SNR error can reach a maximum of 1.2 dB and an average of 0.7 dB. We also notice that the impact of the NF parameter is a linear curve. The slope of this curve is proportional to the amount of uncertainty to add to the NF parameter.

3.5.6 Fiber-Related Parameters

The fiber-related parameters model the various effects that occur in the fiber. The span loss is the main impairment that greatly affects the quality of the optical signal. It could be caused by the attenuation coefficient of the fiber (also called lineic loss) and the fused losses such as splices (which are more numerous in older fibers) and the connectors losses.

The span loss is characterized in GNPy by the attenuation coefficient α , the connector losses (CONN_IN and CONN_OUT) and the splice losses called fused in the tool (FL for fused loss). α depends on the type of fiber but can have a small uncertainty due to manufacturing. Although fiber type has been the subject of some papers in the literature [86], operational experts confirm that these data are generally available with low degree of uncertainty. The total span loss can be calculated if the total power is measured from the input and the output of the span.

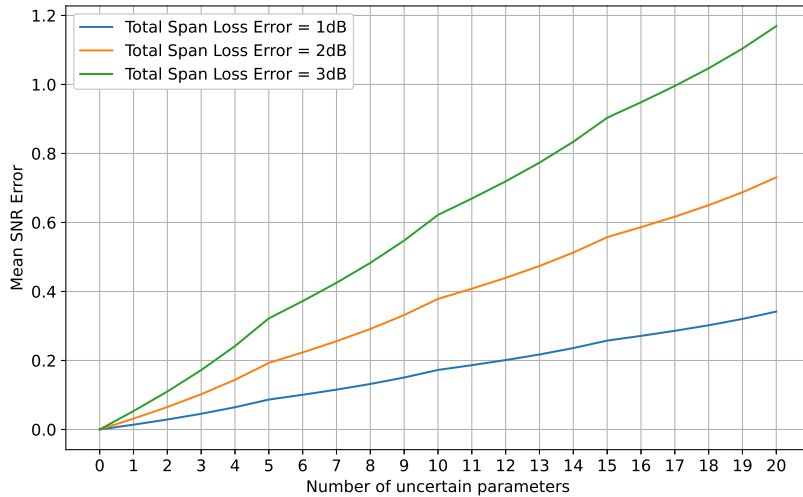


Figure 3.8: Mean SNR error generated by gradually changing the CONN_OUT loss in each span in the 4 OMS FLAT_NET topology in power mode

If we assume that these measurements are not available, the impact depends on the amplifier operation mode as shown in Figures 3.8 and 3.9. Here, we only change the CONN_OUT parameter to increase the total loss. In power mode, the amplifier increases the gain to maintain the target output power which increases the generated ASE noise. The SNR error generated is similar to the NF uncertainty case, and can reach up to 1.2 dB on average. In gain mode, the amplifier does not change its gain, so the power level gradually decreases through the line which results in a poor SNR. The SNR error here increases

exponentially with the CONN_OUT error and can reach 7 dB in the worst case.

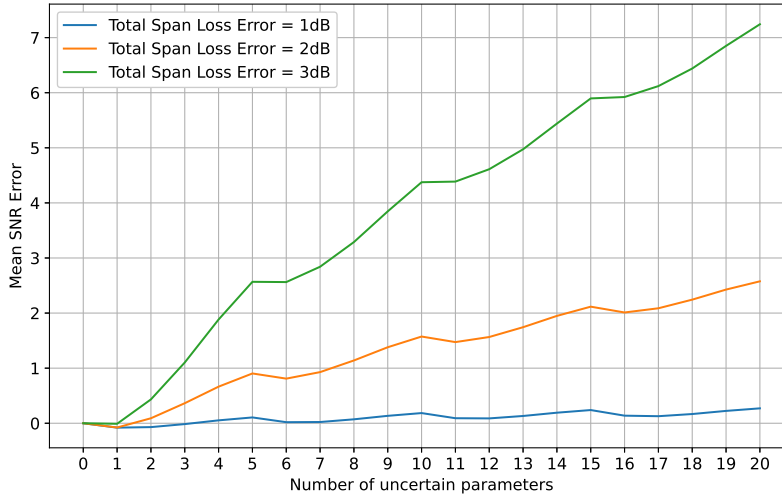


Figure 3.9: Mean SNR error generated by gradually changing the CONN_OUT loss in each span in the 4 OMS FLAT_NET topology in gain mode

Besides the fiber attenuation, GNP_y takes into account the lumped loss on the connectors (in and out) and splices. The distribution of this kind of attenuation is important because it affects the power injected into the fiber, which affects the intensity of the nonlinear effects, and the SRS shape applied to the spectrum. In Figure 3.10, we change the CONN_IN parameter between 0.1 and 2 dB, while conserving the total loss in each span in the first OMS of Figure 3.5. We also change the power per channel at the output of the amplifier. We notice that the SNR spectrum changes drastically based on the injected power in the fiber. At 4 dBm, the nonlinear effects are less prominent so the impact of the connection loss only reaches up to 1.6 dB for the worst channel in the spectrum. But at higher power, large CONN_IN values reduce the injected power into the fiber, which reduces the SRS effects which leads to up to 8 dB error in the worst channels.

`./chap_fig/_chap2/CON_in_SRS_impact-eps-converted-to.pdf`

Figure 3.10: SNR spectrum generated by changing the CONN_IN loss in each span in the 1 OMS FLAT_NET topology

Based on these observations, we chose two sets of parameters to model the uncertainty on the fiber loss distribution. If the total span loss can be measured through power mea-

measurements, we calculate the residual loss (total span loss - (lineic loss + VOA losses)), then distribute the residual loss on the CONN_IN and CONN_OUT parameters. If the total span loss is unknown, we consider each source of potential loss other than the lineic loss as uncertain. In the case of GNPY, these parameters would be represented by the CONN_IN, CONN_OUT and losses due to fiber fusing (FL).

3.5.7 Optical Power

Optical power can be reliably measured at different points in the field. But, this is highly dependent on the operator and vendor specifications. Some equipment boards have optical power monitoring (OPM) units that can measure the signal spectrum, while other boards contain photo-diodes that can only measure the total power. We considered three different scenarios related to power measurements: i) no power measurements are available, ii) total input and output power can be measured on each amplifier and the power spectrum is available at the output of the booster and the input of the preamplifier and iii) power spectrum can be measured at each point of the OMS. The last assumption is currently unrealistic, but it could exist in future operational systems thanks to technological advances like the use of ML techniques as proposed in [63].

3.5.8 QoT metrics

As discussed in chapter 2, we consider that the only QoT indicator that can reliably be measured in the field is the BER, because it is reported by the transponders. OSNR can technically be measured by using an Optical Spectrum Analyzer (OSA), but it requires shutting off services in order to measure the ASE under the channel, which is not possible in the field. The SNR however can be calculated from the BER if the back-to-back characterization of the transponder is known. As GNPY is able to calculate the SNR and the OSNR of each service, the QoT metric we rely on is the SNR.

3.5.9 Discussion on parameters uncertainty

We summarize the analysis of the parameter uncertainty in Table ???. We specify in the "GNPY" column whether the parameter is required by the current version of the GNPY tool. We omit the parameters that are not required from our study. We consider that the amplifier NF and the fiber loss distribution modeled using the CONN_IN, CONN_OUT and FL are the main uncertain parameters that impact QoT estimation. The fiber losses mainly affect the nonlinear effects in the fiber by attenuating the power injected into the fiber. The NF affects the linear noise by injecting ASE noise in the transmission. This means that using these two parameters we can compensate for any inaccuracy in the linear

and nonlinear noise contribution in the SNR. We can potentially have other parameters whose value is uncertain, or simply incorrectly saved in the control plane. However, if we consider a wider range of uncertain parameters, the number of possibilities increases and it becomes hard to find the correct set of parameters. Furthermore, we only consider these parameters as a single mean value over all the channels in the transmission. The potential per channel variation of the NF and loss parameters will be addressed in our learning processes. Lastly, these are the parameters that we consider uncertain by default. If we have any knowledge that an additional parameter is uncertain in a specific network scenarios, this parameter must be taken into consideration. For this reason, the learning processes that we set up can be easily modified to account for a different set of uncertain parameters.

3.6 Conclusion

In this chapter, we presented the framework that describes the use of QoT estimation in a real network scenario, and the process by which we can improve the accuracy of this estimation. We consider that parameter uncertainty is the main factor that affect QoT estimation, and propose to use learning processes based on AI techniques to reduce the effects of this uncertainty on the QoT. In this chapter, we presented two elements of this framework which are the QoT models and the network state. We presented two QoT estimation tools: an analytical tool (GNPy) and a ML-based one (Neural Network). We then presented the choice of parameters that constitute the network state and discussed their relative degree of certainty. In chapter 4, we present the learning processes that aim to improve the QoT estimation.

	Parameter/QoT indicator	GNPy	Certainty and Assumption
TRX	Service configuration (Bit rate, Baud Rate, Spacing ...)	Individual service parameters	Accurate: can be obtained from the controller
	Back-to-back penalty	OSNR penalty	We assume that TRX characterization is available
	Bit Error Rate (BER)	Not Computed	Certain: assumed to be always measured in the field
	Signal to noise ratio (SNR)	Computed	Certain: deduced from the BER
ROADM	Equalization	Fixed PCH or PSD	Equalization information should be available on the controller
	WSS configuration (signal loss, filtering penalties ...)	Not required	Not studied
EDFA	Noise Figure	Required	Uncertain: usually missing or not available in the field
	Total Gain	Required	The total gain value is known and the gain spectrum assumed to be flat tilted. Potential uncertainty on gain ripple
	Total input power	Required	Accurate but depends on OPM availability
	Total output power	Required	Accurate but depends on OPM availability
	Per channel input/output power	Required	Accurate but depends on OPM availability
Fiber	Total loss	Required	Depends on available power measurements
	Loss distribution	Required	Uncertain and modeled as fused and connection losses
	Fiber coefficient (attenuation, dispersion, gamma, PDL)	Required	Assumed to be certain

Table 3.2: Summary of the parameters constituting the GNPy based network state and their level of certainty

AI Techniques Comparison Using Synthetic Data

Contents

4.1	Introduction	64
4.2	Problem Formulation	64
4.3	Simulation Network Topology	65
4.4	First Approach: Optimizing Network Parameters	66
4.4.1	LP1 - Bayesian Optimization for Analytical QoT Model	67
4.4.2	LP2 - Gradient Descent for ML-based QoT Model	76
4.5	Second Approach: Adapting the QoT Model	84
4.5.1	LP3 - Transfer Learning to Adapt a ML-Based QoT Model	84
4.6	Third Approach: Detecting Parameter Changes	92
4.6.1	LP4 - Detecting parameter changes using reinforcement learning	94
4.6.2	Discussion Around LP4	97
4.7	Conclusion	98

4.1 Introduction

We proposed in the previous chapter a framework based on a closed-loop that uses QoT measurements collected by the controller to improve the QoT estimation. In this chapter, we propose several approaches that can achieve this improvement, which constitutes the feedback part of the loop. We distinguish three different approaches: i) optimizing network parameters, ii) adapting the QoT model to parameter uncertainty and iii) detecting parameter changes. For each approach, we propose one Learning Process (LP) whose performance is assessed using several simulation scenarios.

The rest of this chapter is organized as follows. In Section 4.2, we present a mathematical formulation of the parameters refinement problem. In Section 4.3, we show the two network topologies we used in our simulations in order to test the LPs we propose. Then, the next three sections (4.4, 4.5 and 4.6) are dedicated to present the three approaches mentioned above. In each one, we first present the concept of the learning process, then the ML algorithm used, and finally the simulation results.

4.2 Problem Formulation

The problem we seek to solve is how to utilize the data measurement collected by the controller to improve the accuracy of the QoT estimation using an analytical model or a machine learning model. We can represent our network as a set of nodes N and connections C where a set of lightpaths $L = \{l_i : i \in [1, N_L]\}$ are deployed. The network state P is the sum of all parameters that describe the network. We split the parameters \mathcal{P} into certain \mathcal{P}_C , which represent parameters whose values are trusted, and uncertain parameters \mathcal{P}_U , which represent parameters whose values might be missing or have changed during the life-cycle of the network. Furthermore, each lightpath l_i has a set of parameters X_i which include the transponder configuration (bit rate, modulation format ...) and the route of the lightpath.

The list of QoT indicators that can be measured in the field is denoted as Q^j . We assume that for each lightpath we can have a set of QoT indicators $Q = \{Q^j : j \in [1, N_Q]\}$ noted as Q_i for measured QoT and \tilde{Q}_i for estimated QoT. The estimated QoT for each lightpath depends on X_i , P_C and P_U .

We represent the analytical tool estimation by a parametric function f such as $\tilde{Q}_{l_i} = f_{P_U}(X_i)$ with only P_U as function parameters since certain parameters are part of the behaviour of the function. The problem we try to solve is how to set up a learning process that takes as input the measurements Q , the parameters P_U to improve the QoT estimation \tilde{Q} . Improving the QoT estimation means reducing the estimation error which is the difference between \tilde{Q} and Q . This can be solved in three potential manners.

Firstly, we can change the values of the uncertain parameters P_U so as the estimation

error is reduced. We call this approach *Uncertain Parameter Optimization*. In Section 4.4, we propose two learning processes that utilize this approach. Each learning process is intended for a different category of QoT estimation model: analytic and ML-based.

Secondly, we can modify the QoT model itself in order to be adapted to the QoT measurements. We call this approach *Adapting the QoT model to Parameter Uncertainty*. We propose a learning process intended for ML-based QoT estimation model. This approach is expanded upon in Section 4.5.

In the third approach, we aim to detect any changes in the performance of the network and ascertain the source of this change. This allows us to keep an up-to-date network state in real time, and avoid large variation in QoT estimation. We call this approach *Detecting Parameter Changes* and propose a learning process based on Reinforcement Learning (RL) algorithms in Section 4.6 to perform it.

4.3 Simulation Network Topology

In order to evaluate the performance of aforementioned learning processes, we set up by simulation a network topology consisting of a flat transmission line as depicted in Figure 4.1 that we refer to as (FLAT_NET).

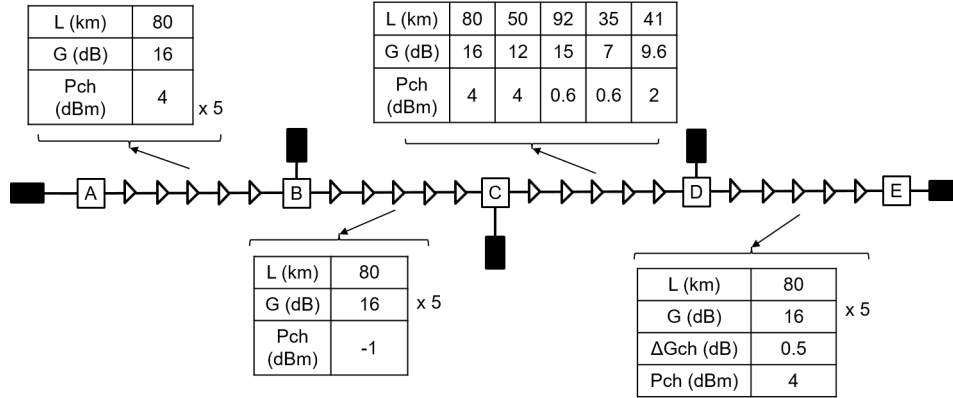


Figure 4.1: The FLAT_NET simulation topology

FLAT_NET is a uni-directional transmission line composed of 4 OMS (Optical Multiplex Section) with 5 spans each. The span configuration is different in each OMS as detailed in the in Figure 4.1. L represents the length of the span. All fibers are SMF with 0.2 dB/km attenuation coefficient. The G represents the gain of the amplifiers. The gain spectrum of all the amplifiers is tilted by 2 dB with no ripple, except in the case of the OMS between nodes D and E where we add a ripple by adding a small uncertainty ΔG_{pch} on the per channel gain value of each amplifier. This uncertainty is sampled from

a normal distribution with a 0.5 standard deviation. Finally, we set a per channel power target P_{ch} at the output of each amplifier, which corresponds to the output power of the central wavelength in the spectrum. The amplifier gains are set to compensate the fiber attenuation, and ensure this power target. We do not consider the amplifier limitations in terms of minimal input/output power and maximum/minimum gain.

The OMS configuration in this simulation study is set so that we can test the performance of the lightpaths in different OMS scenarios. The span configuration of the two OMSs AB and BC are similar with a 80 km length and 16 dB gain. However, in AB the power target is set to 4 dBm which puts the transmission in the non-linear regime, while BC has a power target of -1 dBm, which means the non-linear effects are greatly reduced. This difference in configuration allows us to test the performance in different transmission regimes. In the OMS CD, we set a different configuration for each span in order to test whether this variation in configuration affects the learning processes or not.

Using this topology, we simulate a full load transmission between each two adjacent nodes. The spectrum grid is composed of 60 channels with 100 Ghz spacing and 90 GBaud services. We do not consider the mixed rate scenario in our study. All the channels are launched with the same power, and at the output of each ROADM, we equalize the signal to a flat spectrum with a total power of -15 dBm.

4.4 First Approach: Optimizing Network Parameters

As explained in Section 4.2, we can improve the QoT estimation if we reduce the error between the QoT measurements Q and the QoT computed by the parametric function representing the QoT tool $\tilde{Q}_{l_i} = f_{P_U}(X_i)$. We can assume that the correct values of the uncertain parameters P_U are the ones which lead to an estimated QoT as close as possible to the measured values. Therefore, we can consider the problem as a general optimization problem in order to find the set of values $V_{U^*} = \operatorname{argmin}_{V_U} \{E_{l_i} = |Q_{l_i} - f_{V_U}(l_i)| : i \in [1, M]\}$. We can use an optimization algorithm that takes as objective function the aggregation of the difference between measured and estimated QoT indicators for each lightpath in the network (i.e., *the estimation error*), and looks for the values of the uncertain parameters P_U that minimize this function.

The algorithm is guaranteed to converge to an optimum as long as all the inaccuracy of the QoT estimation can be explained by the selected uncertain parameters. Actually, the error in estimation is not only due to the imprecision of the input parameters but also to the imperfection of the analytical model that does not perfectly match with the real field (e.g., temperature, unclean fiber connector). Consequently, we do not expect to converge towards the real physical values of the uncertain parameters. We can improve the QoT

estimation as long as we get a set of values, even if it is different from the real one, that compensate for the error between the measured and estimated QoT. Nevertheless, values of uncertain parameters should be as close as possible to the reality in order to guarantee an accurate estimation of QoT for the long paths as well as the short ones. Therefore, the ranges of values of each uncertain parameter should be specified in the optimization process and the objective function should be enriched with several QoT metrics and indicators such as the optical power in specific points of the transmission line.

This parameter optimization approach can make abstraction of the QoT tool used by considering the function f as a black box function. However, we distinguish two categories of QoT computation tools. The first category comprises tools based on an analytical model and the second category comprises tools based on a ML model. For each category, we propose one LP. Although each LP has a different optimization algorithm, these two LPs use a similar processes to choose the set of uncertain parameters. The first learning process is presented in Section 4.4.1, while the second one is presented in Section 4.4.2. The first simulation results using LP1 were published in a conference paper [17].

4.4.1 LP1 - Bayesian Optimization for Analytical QoT Model

The resolution of the problem formulated above could be done using multiple optimization such as genetic algorithms and gradient descent. However, in this learning process, we focus on the Bayesian Optimization (BO) algorithm. BO is a heuristic search-based optimization algorithm that can converge towards an optimum in a low number of iterations, and requires minimal fine-tuning. It incorporates prior belief about a function f and updates the prior with samples drawn from f to get a posterior that better approximates f . The model used for approximating the objective function is called *surrogate model*. Bayesian optimization also uses an acquisition function that directs sampling to areas where an improvement over the current best observation is likely. The iteration process of the algorithm is represented in Figure 4.2. We use BO with a Gaussian Process as the surrogate function and Maximum Expectation as the acquisition function. No further fine tuning or improvement over the base algorithm is explored, as the performance is satisfying.

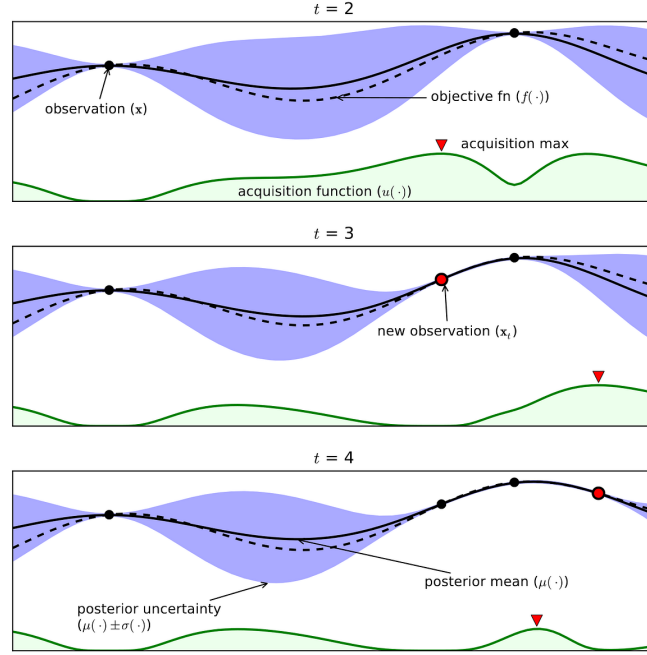


Figure 4.2: Bayesian optimization algorithm process. Figure taken from [87]

As explained above, the BO algorithm makes abstraction of the physical problem and simply selects a list of unknown parameters, an objective function, and a list of constraints in order to launch the optimization. For the purpose of testing our learning process in a simulation environment, we set up the following steps:

1. Optimization configuration

- (a) **Choose a subset of the topology:** We perform the optimization per OMS unless otherwise noted.
- (b) **Select the unknown parameters:** We consider two types of parameters: the amplifier noise Figure and the connection/lumped losses.
- (c) **Select the measurements in the objective functions:** We use the SNR spectrum at the reception and/or the pre-amplifier output power spectrum as measurements depending on the network scenario. The objective function consists of the Root Mean Squared Error (RMSE) of each individual measurement normalized in the $[0, 1]$ range.

2. Running the optimization

- (a) **Sample the initial state:** It represents the current state of the network which corresponds to the non accurate values of the uncertain parameters. We sample

these values using a uniform distribution from a specific range per parameter type (NF: [4, 9] and loss [0, 3]).

- (b) **Sample the target state:** It is the accurate values of the uncertain parameters. We sample these values in the same way as the initial ones.
- (c) **Calculate the target QoT indicators:** We simulate with GNPpy the transmission using the target state parameters then calculate the performance indicators. We refer to it as *Target QoT*.
- (d) **Run the optimization loop:** We set the current state to the initial one. Then, in each iteration, we calculate the performance and compare it to the target performance in order to evaluate the objective function. Depending on the obtained result, a new set of parameters is proposed. The loop ends when the maximum number of iterations is met.

3. Optimization results and validation

After performing the optimization, we select the set of parameters that gives the best results, hereafter called the *best state*. Then, we focus on metrics related to SNR as listed below to evaluate the performance:

- (a) **The residual SNR error:** We simulate the SNR spectrum using the parameters values of the best state obtained in the simulation. Then, we calculate the error between the SNR spectrum of the target state and the best state.
- (b) **The residual error in the parameters:** We calculate the difference between the best value and the target value for each parameter. This error technically cannot be measured in a real scenario since we do not have the target parameter values. But, we compute it in simulation in order to evaluate if the algorithm converged to the right parameters state or an alternative one.
- (c) **The residual redesign SNR error:** We calculate the residual SNR as explained above but after increasing the power target of each amplifier by 2 dB. This metric allows us to validate the parameter state that we obtained by testing if it reduces SNR error in different network configurations.

This set of metrics permits to evaluate the optimum that the algorithm has reached. The goal of the parameters optimization use case is to improve the SNR estimation in the case of future lightpath deployment requests. Therefore, we need to test the obtained parameters with new network configurations. However, since we study in this section the transmission with a fully loaded spectrum, we cannot add new lightpaths. Another method to evaluate our solution consists of evaluating the performance metric

with a modified set of certain parameters. For this purpose, we choose to increase the target output power at every amplifier by 2 dB. Then, we calculate the difference in performance using the target and the best state. We call this validation metric the "*redesign SNR error*".

Using this learning process, we run multiple experiments. As explained in Section 3.5, we chose the amplifier noise Figure and the lumped loss distribution as uncertain parameters. The noise Figure is always modeled using a single mean value for each amplifier. However, the lumped loss can either be modeled using a single connection parameter in every span, or we can consider the connection in and out of the fiber as well as all the fused losses in the span. This depends on the level of knowledge of the power in and out of the amplifiers. Furthermore, we consider that it is possible that the gain ripple of the amplifiers is uncertain. However, we do not consider the per channel gain values as uncertain parameter, since it would lead to a very large number of parameters to optimize. In order to test the LP performance in each case, we propose the following simulation scenarios:

4.4.1.1 Simulation Scenario 1: Optimization Without Power Measurements

If no power measurements are available, the total span loss cannot be measured. This means that we need to consider all the sources of fiber loss distribution (CONN_IN, CONN_OUT and FUSED). We suppose that each loss parameter can vary from 0 to 1.5 dB, and we assume that the number of fused in the span are known. The noise Figure of each amplifier is also optimized. We use the SNR spectrum to calculate the objective function. We evaluate the performance of the LP using the aforementioned metrics. We run the simulation with the following configuration:

- Network topology: separate optimization for each OMS: AB, BC, CD and DE in the FLAT_NET network topology
- Uncertain parameters: 6 NF parameters, 5 CONN_IN, 5 CONN_OUT loss and 3 FL for each span.
- Objective function: RMSE(target SNR spectrum, current SNR spectrum)

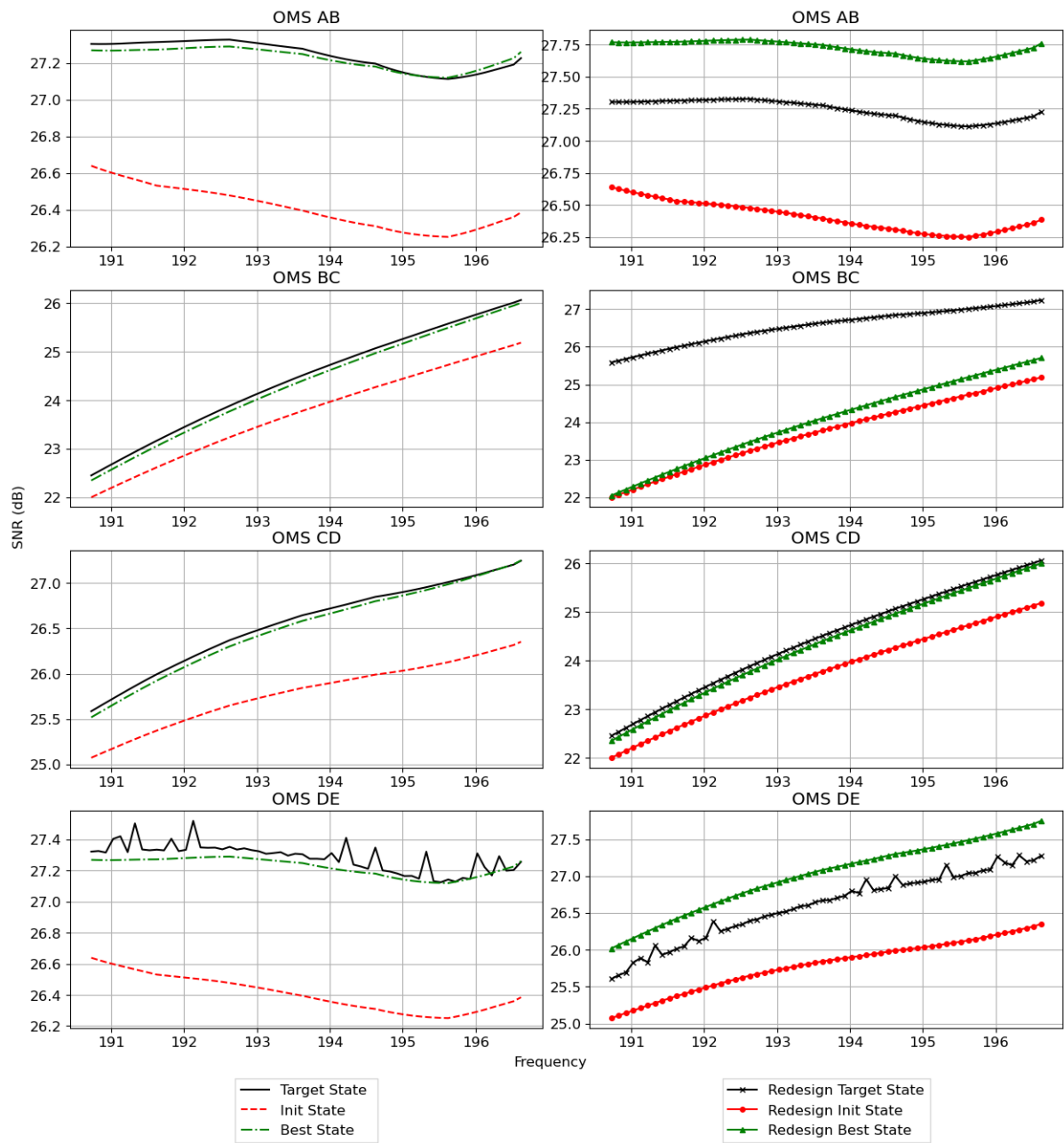


Figure 4.3: LP1 - Simulation 1 Results: SNR spectrum in each network state during the learning process

We notice in Figure 4.3 that the LP can reduce the mean SNR error from more than 1 dB to 0.1 dB in all cases. The residual error is very minimal, except in the case of OMS DE due to the uncertainty on the gain spectrum. However, in Figure 4.4, we notice that the error

on the parameters is large (up to 1.5 dB) which means that the set of parameters to which the LP converges is different from the target one. In OMS CD, the span configuration has a lot of variation because it was generated randomly, so we expect that the LP will converge to the correct parameters. However, we do not observe any gain in terms of parameter error from OMS CD.

The redesign SNR error in 4.3 is quite substantial (mean error of 1.2 dB). This means that the best state parameter value that we obtained does not fit the real physical state of the network. This can be explained by the fact that the SNR error can be both compensated using the fiber loss parameters and the NF, since the SNR calculations are based on the GNLI model. Furthermore, the impact of the fiber loss parameters is much higher than the NF in this case as seen in Section 3.5. Both of these factors mean that the LP can converge to a large set of alternative solutions that minimize the SNR of the initial design, but not the redesign SNR.

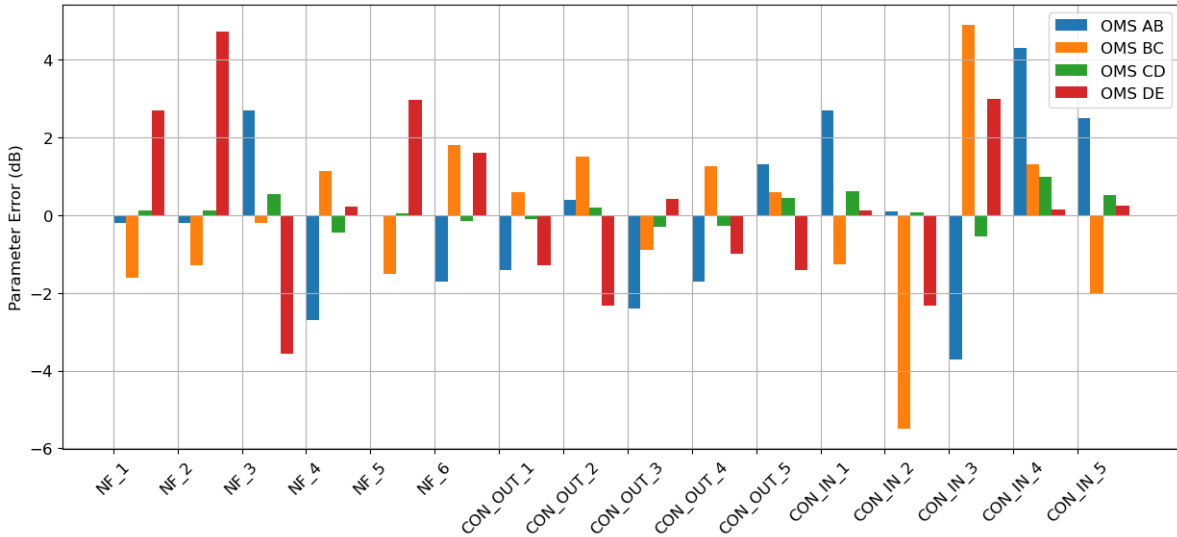


Figure 4.4: LP1 - Simulation 1 Results: Error between the target and best values for each uncertain parameter in the learning process

4.4.1.2 Simulation Scenario 2: Optimization With Power Measurements

In this scenario, we assume that we can measure the power of the spectrum at the input of the booster and at the output of the pre-amplifier, and that we can measure the total power at the output and input of every amplifier. This leads to two main changes in the learning process. First, we can now calculate the total span loss, which means that we can model the fiber loss distribution using the CONN_IN parameter only, as discussed in Section 3.5. Secondly, we can use the power measurements at the end of the OMS as an

objective function in the optimization.

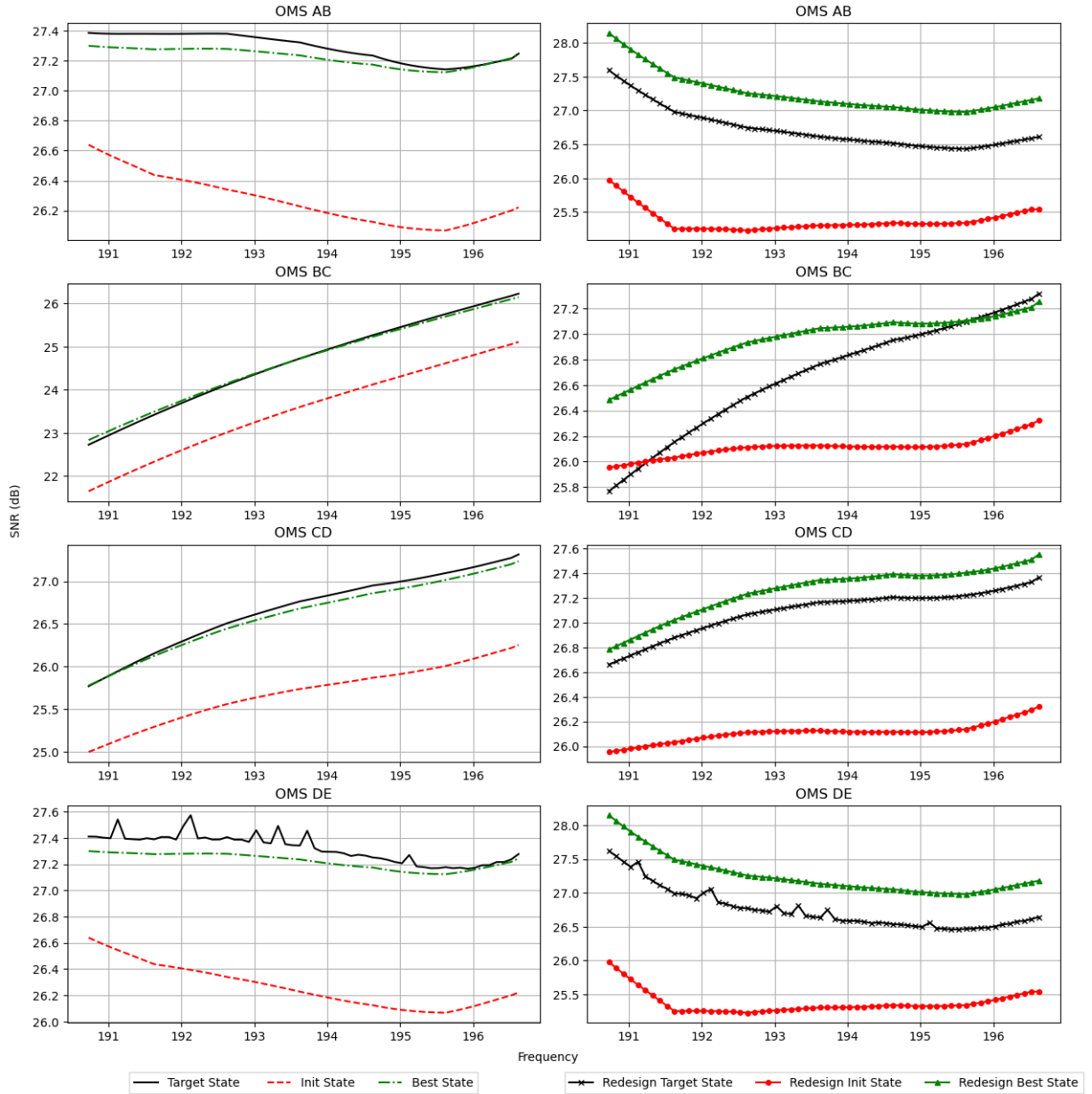


Figure 4.5: LP1 - simulation 2 results: SNR spectrum in each network state during the learning process

- Network topology: separate optimization for each OMS: AB, BC, CD and DE in the FLAT_NET topology
- Uncertain parameters: 5 NF parameters and 5 CONN_IN.

- Objective function: $\text{RMSE}(\text{target SNR spectrum, current SNR spectrum}) + \text{RMSE}(\text{target PWR spectrum, current PWR spectrum})$

Figure 4.5 gives the residual SNR error in function of the power spectrum for both initial design and the modified design. We notice that the residual error after optimization is below 0.08 dB for SNR. In general the performance is similar for all the OMSs except for the OMS DE due to the gain spectrum uncertainty. In terms of redesign SNR error, the results of this simulation are much better than the previous one (0.7 dB mean error). This can be explained by the fact that the power spectrum forces the algorithm to select parameters that fit the real physical state of the network. In particular, we notice that the OMS DE has 0.2 dB as redesign SNR error.

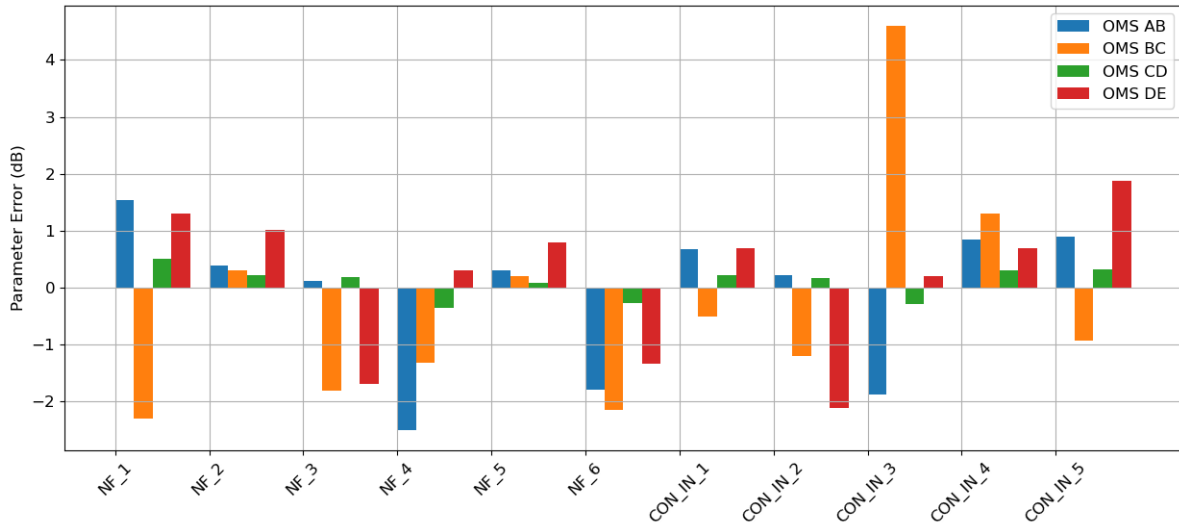


Figure 4.6: LP1 - simulation 2 results: Error between the target and best values for each uncertain parameter in the learning process

The parameter errors in Figure 4.6 seem to be improved compared to the previous simulation. In OMS AB, BC and DE, the mean parameters error is below 0.6 dB. This can be explained by the fact that the NF and CONN_IN impact in different ways the power and the SNR, which means that the consideration of power in the objective function can serve to separate between these parameters. Additionally, we do notice a much greater improvement in OMS CD due to the different span configurations with 0.1 dB error in parameters. Despite of this improvement, we notice that even with the addition of power spectrum, the LP is not guaranteed to converge to the target parameters because the conditions of OMS CD are very special and in an operational network scenario the span configuration is more consistent in terms of target output power.

4.4.1.3 Simulation Scenario 3: Optimization with Power Measurements and Signal Ripple Compensation

In the previous scenarios, we used the NF and fiber loss parameters to compensate the error in SNR and power spectrum estimation. For both parameters, we use the mean value over the spectrum. However it is known that the values of these two parameters depend on the channel index. Furthermore, in an operational network, we can have per channel variation in performance that is not explained by the SRS effect or the gain tilt, such as the gain ripple uncertainty that we introduced in the OMS DE. The two uncertain parameters we chose cannot compensate for this variation at the channel granularity. For this purpose, we extend our LP by adding a second optimization step. In this step, we try to compensate the per channel variation by optimizing a fictional set of parameters ΔP_{ch} , which is an uncertainty on the power per channel at the input of each booster. These parameters are optimized using both the SNR and the power as objective functions, right after performing the optimizing of the NF and CONN_IN parameters. In this simulation, we consider the results of the previous simulation as an initial state, then perform this extra step in the learning process to evaluate the gain in performance using this technique. To simulate the per channel performance variation, we use OMS DE that has an uncertainty on the gain ripple.

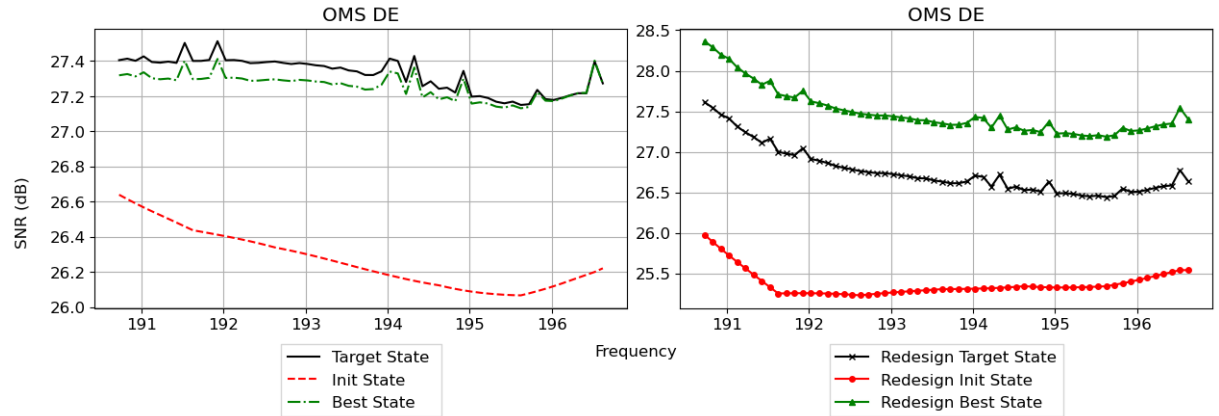


Figure 4.7: LP1 - simulation 3 results: SNR spectrum for each network state during the learning process

- Network topology: OMS DE
- Initial state: Best state from Simulation scenario 2
- Uncertain parameters: ΔP_{ch} power shift for each channel in the spectrum (60 parameters).

- Objective function: RMSE (target SNR spectrum, current SNR spectrum) + RMSE (target PWR spectrum, current PWR spectrum)

In Figure 4.7, we plot the different SNR spectra before and after the second optimization step. We notice a clear improvement of 0.2 dB and 0.3 dB in SNR and power respectively. However, the redesign SNR decreased by 0.1 dB in average. This means that the second step allows us to fit the performance of the transmission line much better when we consider channel dependant performance variation. But, since we compensate the error in performance with a fictional parameter, the resulting network state does not reflect the real network physical state. The power shift that we add compensates the uncertainty in the gain ripple at the end of the OMS instead of after every amplifier. This means that the power spectrum at the output of each amplifier is not accurate, which explains the difference in QoT estimation. An alternative to this power shift could be to consider the gain spectrum in every amplifier as uncertain then optimize it. However this would lead to a similar result, as the objective function is composed of the SNR and power spectrum at the end of the OMS, so we still would not have the accurate power spectrum after each amplifier. Additionally this would add too many parameters in the optimization.

4.4.2 LP2 - Gradient Descent for ML-based QoT Model

The LP2 is intended to refine uncertain parameters in the case where the QoT estimation is performed based on the NN model described in Section 3.4. In the following simulation scenarios, we use the same learning process and evaluation metrics as LP1 in Section 4.4.1. This means that we sample the target and initial state the same way, and we chose the uncertain parameters and objective functions the same way. The only difference is the QoT model we use for QoT estimation, and the optimization algorithm. However, the NN-based QoT tool we trained is not as comprehensive as the GNPpy tool as explained in Section 3.4, which means that the choice of simulation scenarios is more restricted. For instance, we only test this LP in the case where the power measurements are available because the model we trained requires the power spectrum at the input of the OMS. We also run the optimization over two steps. The first step optimizes the NF and CONN_IN parameters, and the second step optimizes the uncertainty on the power spectrum. In this learning process, we use the the Gradient Descent (GD) as optimization algorithm.

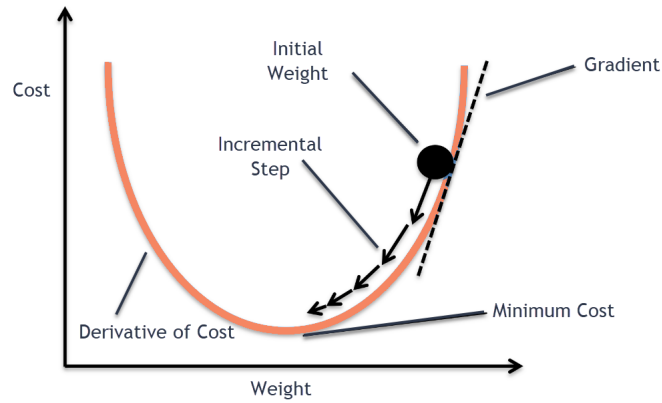


Figure 4.8: Gradient descent algorithm process

GD is a gradient based iterative optimization algorithm. It works by calculating the first order gradient of the objective function in respect of each parameter in the optimization, and gradually changing the parameter by a value proportional to the gradient as shown in Figure 4.8. It is widely used in ML to train neural networks. Gradient descent performs very well especially if the function is convex, but it requires the function to be differentiable. This makes its application in the case of a black box function complicated, which means that it is not suitable for some analytical models.

In our study, we use Gradient Descent in order to optimize the input parameters of our neural network based-QoT model. Neural networks are trained through back propagation using algorithms based on gradient calculations. This means that we can efficiently apply the gradient descent algorithm to perform parameter optimization. Bayesian optimization can also be used for ML-based QoT computation, however gradient descent is much more optimized for this task.

4.4.2.1 Simulation Scenario 1: Comparing SNR Estimation Using NN and GNP_y

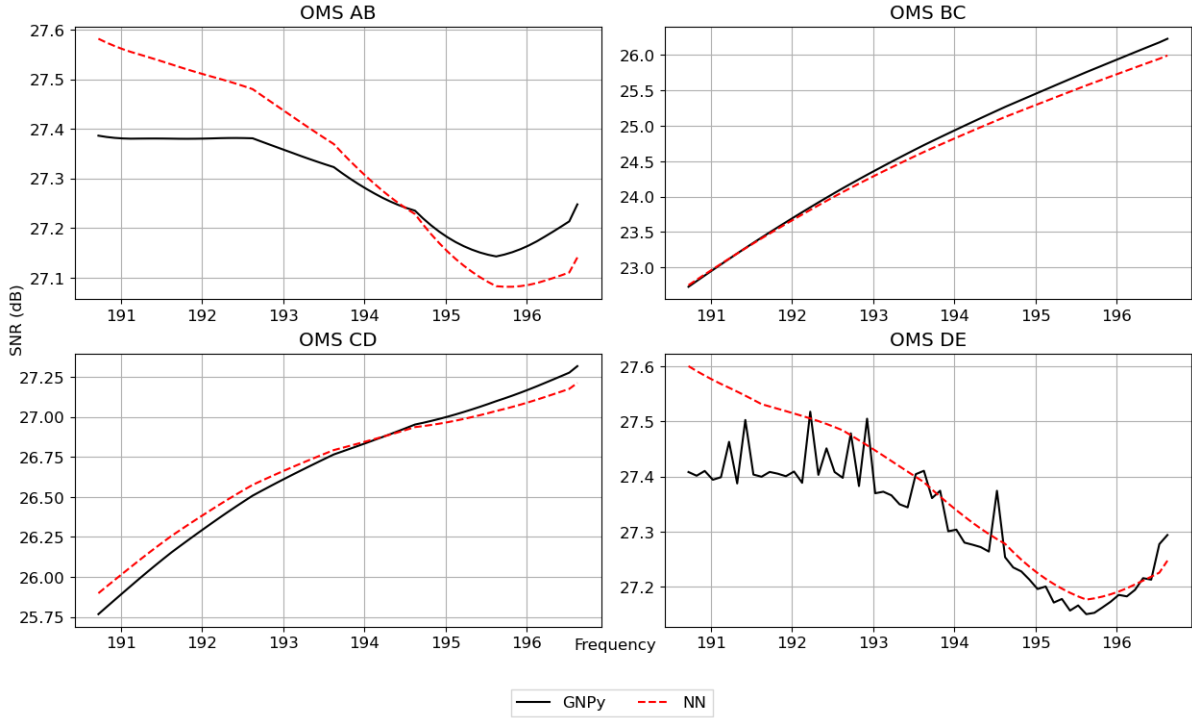


Figure 4.9: LP2 - simulation 1 results: comparison between the SNR estimation of GNP_y and that of NN

We start our study by comparing the performance of the NN model to GNP_y estimation. In Figure 4.9, we show the SNR estimation for each OMS of the FLAT_{NET} topology using both models. In OMS AB, we notice that the estimation difference reaches up to 0.2 dB with a mean of 0.08 dB. The shape of the SNR spectrum is also more flat using the NN compared to GNP_y. In OMSs BC and CD, the estimation difference is lower with a mean values of 0.03 dB and a maximum value of 0.18 dB. This can be explained by the fact that the transmission in OMSs BC and CD is in the linear regime, as shown by a positive SNR tilt, while OMS AB has more nonlinear effects in the transmission. Because of this difference, it is easier for the NN model to learn an accurate SNR estimation when there are small impairments in the transmission. We notice the same effect in the OMS DE which is also in the nonlinear regime, characterized by large nonlinear noise due to a higher power at the input of the fiber. Furthermore, another limitation of our NN model is that it does not take into consideration the ripple in the gain spectrum. This means that in our simulations, we must use the GNP_y to simulate the target state of this OMS in order to study the effects

of the gain ripple. Overall, we consider that the performance of the NN model is accurate enough to use it as a QoT model to test our LPs. This does not mean that it can replace GNPpy as a QoT model, since it was trained using data from GNPpy. However the goal of our study is to test the learning processes and not to produce a perfectly accurate NN. In the next two simulations, we will test how the parameters optimization process performs when applied to the NN model.

4.4.2.2 Simulation Scenario 2: Optimization With Power Measurements

We run the first step of the optimization process similarly to the simulation scenario 2 in Section 4.4.1. The target, initial and best states are simulated using the NN model, while the parameter optimization is performed using the GD algorithm. The initial states are the same as we used in the previous simulation. We do not test the optimization on OMS DE because the NN model is not able to simulate amplifier gain ripple.

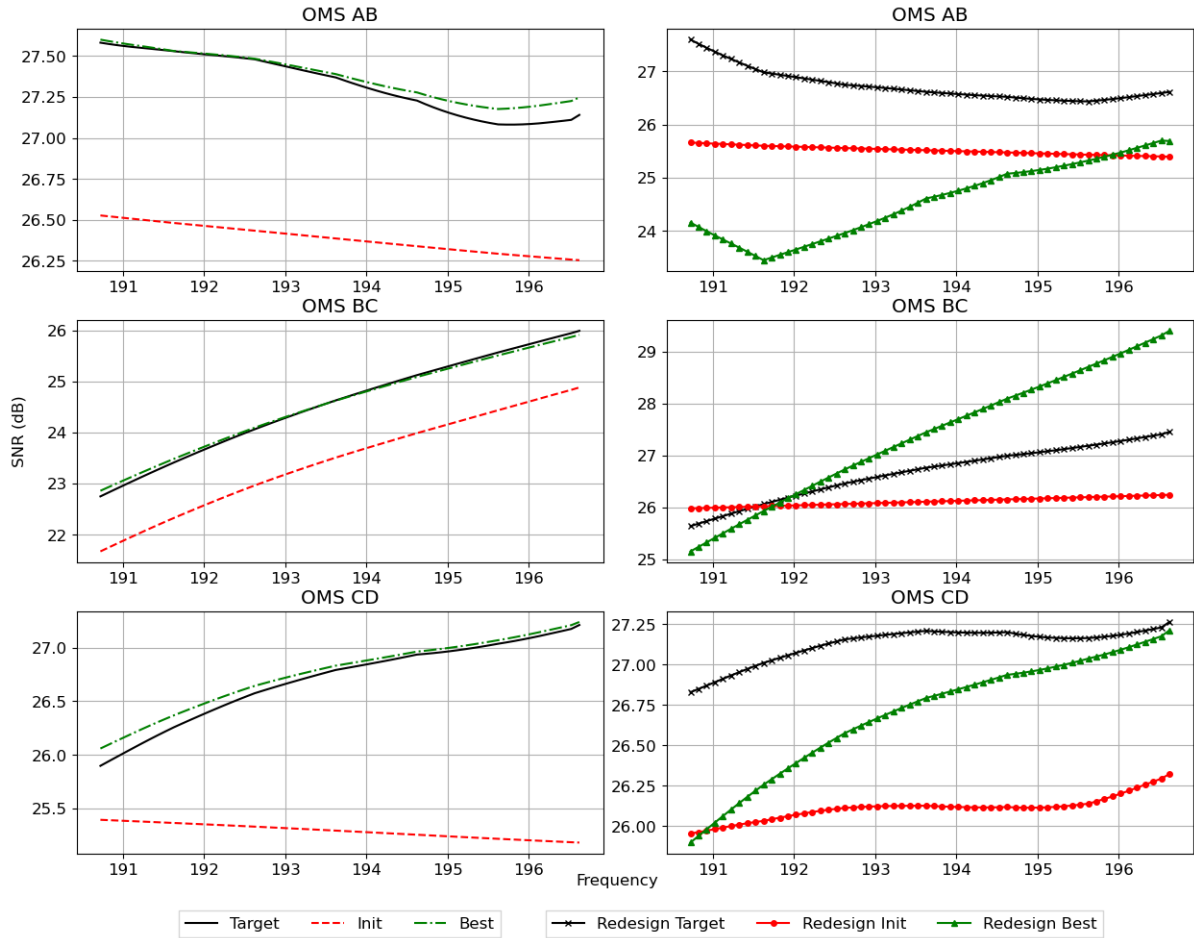


Figure 4.10: LP2 - simulation 2 results: SNR spectrum for each network state during the learning process

- Network topology: separate optimization for each OMS: AB, BC and CD in the FLAT_NET topology
- Uncertain parameters: 5 NF parameters and 5 CONN_IN.
- Objective function: $\text{RMSE}(\text{target SNR spectrum, current SNR spectrum}) + \text{RMSE}(\text{target PWR spectrum, current PWR spectrum})$

We show in Figure 4.10 the SNR estimation results for each OMS. We first notice that the residual SNR error is successfully minimized in each case with a mean value of 0.03 dB and a maximum value of 0.11 dB. The SNR improvement between the initial state and the best state is around 1 dB in all cases. In the OMS CD specifically, we gain 1.5 dB on average, but only 0.5 dB improvement in the worst channel. The redesign residual error, on

the other hand, is much higher with up to 3.5 dB in the OMS AB. Additionally, the shape of the SNR spectrum does not correspond to that of the target in the case of redesign. This means that optimization can accurately match the SNR spectrum in the objective function, but the parameters we obtain do not reflect the real physical state of the network. We noticed the same issue in LP1, however the error in LP2 is much higher meaning this learning process cannot be reliably used to estimate the performance in case of a change in the network configuration. These results can be attributed to the fact that the relationship between the inputs and outputs of the NN model is not based on closed form mathematical formulas and depends on the NN weights that the model converged to. It is possible that an NN model that was trained differently would have better redesign performance, however this was not the case in our model.

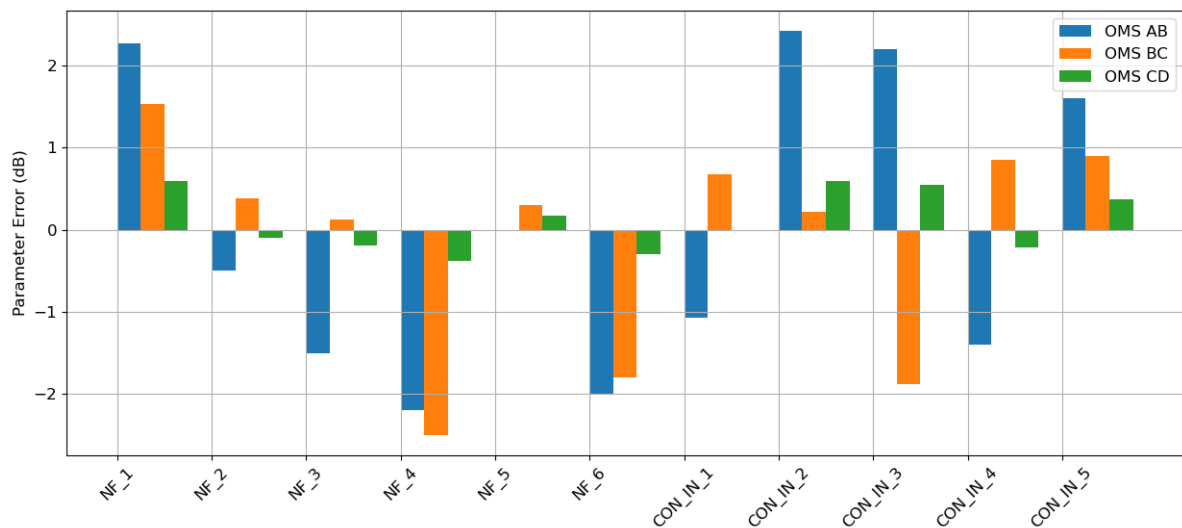


Figure 4.11: LP2 - simulation 2 results: Error between the target and best values for each uncertain parameter in the learning process

In Figure 4.11, we plot, for each OMS, the error in parameters which represents the difference between the target and best parameter value obtained after optimization. We notice that none of the parameters were predicted accurately and that the error can reach up to 2 dB. This matches the results we obtained in LP1 and is due to the fact that multiple alternative parameters values are able to accurately match the target SNR as well as the power spectrum at the end of the OMS.

4.4.2.3 Simulation Scenario 3: Optimization With Power Measurements and Signal Ripple Compensation

In this simulation, we test the ability to compensate channel dependant inaccuracies in the QoT by assuming an uncertainty in the input power spectrum. For this purpose, we use the OMS DE which has inaccuracy on the ripple of the amplifier gain. We must first apply step one (NF and CONN_IN optimization) on this OMS using the GNPpy power and the SNR spectrum as an objective function. This is due to the fact that the NN model cannot simulate the gain ripple inaccuracy. However, if we run the optimization using the QoT estimated by GNPpy, the optimization will compensate both the inaccuracy on the parameters and the difference between the NN and GNPpy. To avoid this case, we run the simulation using the best state obtained in the results of simulation 2 in Section 4.4.1, then optimize the NF and CONN_IN parameters in order to reduce the estimation error between GNPpy and the NN model.

- Network topology: OMS DE
- Initial state: best state from the simulation scenario 2 in Section 4.4.1
- Uncertain parameters: 5 NF parameters and 5 CONN_IN
- Objective function: RMSE (target GNPpy SNR spectrum, current NN SNR spectrum) + RMSE (target GNPpy PWR spectrum, current NN PWR spectrum)

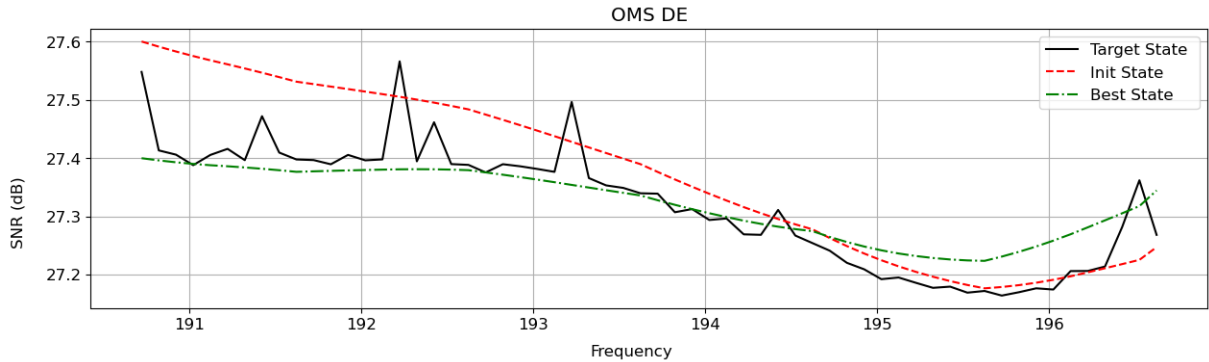


Figure 4.12: LP2 - simulation 3 results: SNR spectrum comparison between GNPpy and NN model before and after the optimization

We show the results of this optimization in Figure 4.12. We notice a small improvement in the mean error from 0.07 dB to 0.04 dB. The improvement is small because the initial error is already small, but this step is necessary to make the NN estimation match GNPpy

as much as possible before optimizing the power spectrum. Then, we run the second step of the optimization in order to compensate for the per-channel variation by optimizing the uncertainty on the power spectrum ΔP_{ch} .

- Network topology: OMS DE.
- Initial state: best state from the previous simulation scenario.
- Uncertain parameters: ΔP_{ch} power shift for each channel in the spectrum (60 parameters).
- Objective function: RMSE (target GNPpy SNR spectrum, current NN SNR spectrum) + RMSE (target GNPpy PWR spectrum, current NN PWR spectrum).

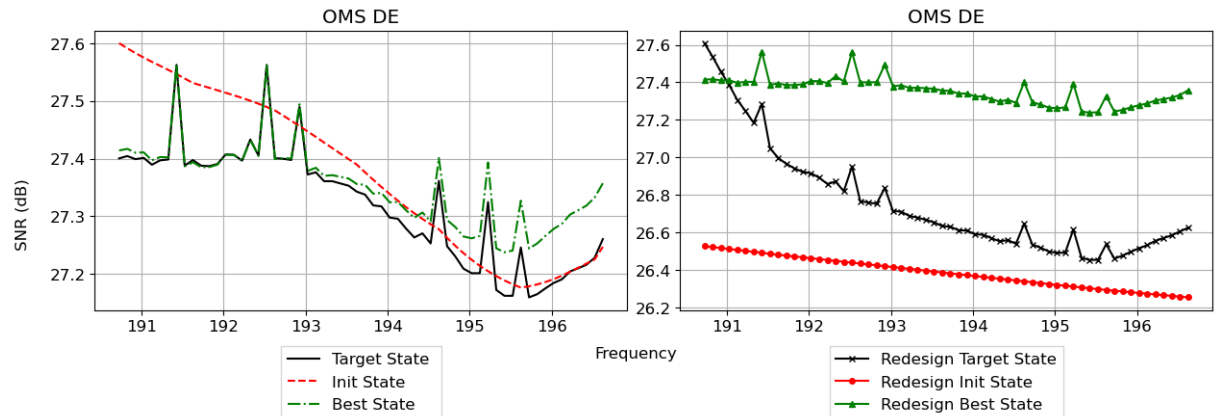


Figure 4.13: LP2 - Simulation 3 results: SNR spectrum for each network state during the power shift optimization

The simulation results are presented in Figure 4.13. We notice that the residual SNR error is not completely minimized as in previous cases. The first half of the spectrum (i.e., frequencies lower than 193 THz) has a low mean error close to 0 dB, but the second half has a higher mean error of 0.042 dB and a maximum of 0.11 dB. We also notice that the shape of the SNR spectrum matches the ripple in the gain spectrum, so this residual error could be due to the remaining inaccuracy on the NF and CONN_IN from the previous optimization as shown in Figure 4.12. Furthermore, the redesign SNR error continues to be significant with 0.7 dB maximum error and 0.38 dB mean error. This was expected as we observed the same pattern when running step two optimization in LP1.

4.5 Second Approach: Adapting the QoT Model

In the previous approach, we changed the input parameters of the QoT model in order to reduce the estimation error. The second approach aims to change the QoT model itself and keep the parameters intact. This does not mean that we attribute the estimation error to the inaccuracy of the QoT model. On the opposite, we continue to assume that some of our network parameters are inaccurate, but we seek to correct this inaccuracy using the QoT model itself. For example, we assume that we only have one uncertain parameter P such as the QoT function is given by $f(P) = SNR$. If the real value of P is 4 such as $f(4) = 15$ and the current value of P is 5 such as $f(5) = 14$, we can solve the problem by adapting the function f into \tilde{f} such as $\tilde{f}(4) = 15$. This adaptation is not always possible especially in the case of analytical models where the QoT estimation depends on closed form equations. For this purpose, we focus in this section on our NN-based QoT model. Indeed, NN-based model are trained using historic data and are not based on empirical equations, which means they can be adapted as long as new data is available that describes the required adjustment in behavior.

To implement this approach, we propose an LP based on Transfer Learning (TL). We surveyed several papers that utilize TL in Section 2.3.4. These papers were trying to retrain a ML model from one network domain to another in order to adapt it. The difference between the domains in these papers is the range or value distributions of the input parameters. In our case, we assume that we have an NN model that was trained on a network with no parameters uncertainty. Then, we retrain this model on a network where the SNR estimation is inaccurate due to parameters uncertainty. We keep the same parameters distributions and we do not introduce any new values that the NN model has never trained on, except for the values excluded in the validation dataset. The challenge is then to change the NN so that its estimation is accurate on the new network while keeping the values of uncertain parameters. In the following section, we present our LP3 and test its performance in two simulation scenarios.

4.5.1 LP3 - Transfer Learning to Adapt a ML-Based QoT Model

The learning process using TL was presented in Figure 2.3. TL is a family of ML techniques used to transfer learned patterns from one ML model to another. Multiple techniques exist to retrain a model especially in the case where retraining data is scarce such as one-shot learning [88]. In this study, we fine tune the initial neural network model by retraining it on a small dataset using the same training techniques as previously presented in Section 3.4. The main factor that we change is the configuration of the retraining dataset. We set up the following simulation steps to test our LP:

1. **Network pre-training:** We utilize the same pre-trained NN as shown in Section 3.4 as basis for each new retraining task
2. **Building the dataset for retraining**
 - (a) **Sample initial state:** It is the known state of the network represented by a set of uncertain parameters values. We sample these values using a uniform distribution from a specific range per parameter (i.e., NF: [4, 9] and loss [0, 3]).
 - (b) **Sample the target state:** It is the accurate set of uncertain parameters values. We sample these values in the same way as the initial ones.
 - (c) **Generate list of network configurations:** Retraining the NN model requires a large dataset. After sampling the initial and target uncertain parameters, we randomly generate a list of values for the certain parameters. Each entry in this list corresponds to a different network configuration but they all assume the same inaccuracy on the uncertain parameters. These parameters will constitute the input features of our dataset.
 - (d) **Simulate the list of network configurations:** We simulate the transmission in each network configuration in order to calculate the corresponding QoT indicators (SNR and power spectrum). We chose to use GNPpy for the simulation in order to add a small inaccuracy in the estimation due to the difference between GNPpy and the NN. These QoT values constitute the output features of our dataset.
 - (e) **Prepare the dataset:** The final dataset is the concatenation of the initial values of the uncertain parameters, the list of network configurations, and the estimated target QoT indicators.
3. **Network retraining:** We run the retraining for a maximum number of iterations (1000 iterations) or until the validation error exceeds a certain threshold
 - (a) **Split the dataset:** We split the dataset into 80% training and 20% validation datasets. We do not employ more advanced training techniques as we assume that the available data for retraining is scarce.
 - (b) **Run the retraining:** We train the model using the new dataset up to a maximum number of iterations, or until the validation error stops improving.

4. Retraining results and validation

After the retraining, we calculate the following metrics to evaluate the performance of the LP:

1. **The size of the training dataset:** We assume that collecting new data for retraining will be hard, since it requires the deployment of different network configurations. The size of the training dataset will be a deciding factor in the feasibility of the LP in a real network scenario.
2. **The training error:** The error is calculated using the RMSE over the SNR spectrum of each network configuration in the training dataset. This allows us to evaluate the performance using the network configurations available for training. This could impact for instance the potential estimation error if we deploy a new service without changing the network configuration.
3. **The validation error:** The model adaptation must make sure that the QoT model remains accurate in a number of different network configurations. We use the validation error to evaluate the new model ability to be generalised. This also replaces the previous redesign technique we used in LP1 and LP2.

We propose to study this learning process in two scenarios. The first scenario is the case where we do not know which parameters are uncertain and the second scenario is the opposite case. The difference between these two cases concerns the structure of the NN model. If we do not know which parameters are uncertain, we must assume that the behaviour of the network has completely changed, and we cannot use the data that the model was initially trained on. If we know which parameters have changed, we can reuse the initial data as long as we specify to the model that the new behavior is due to the change in parameters. We incorporate this difference in our learning process and present the results in the following two sections.

4.5.1.1 Simulation Scenario 1: Retraining the Model Without Knowing Uncertain Parameters

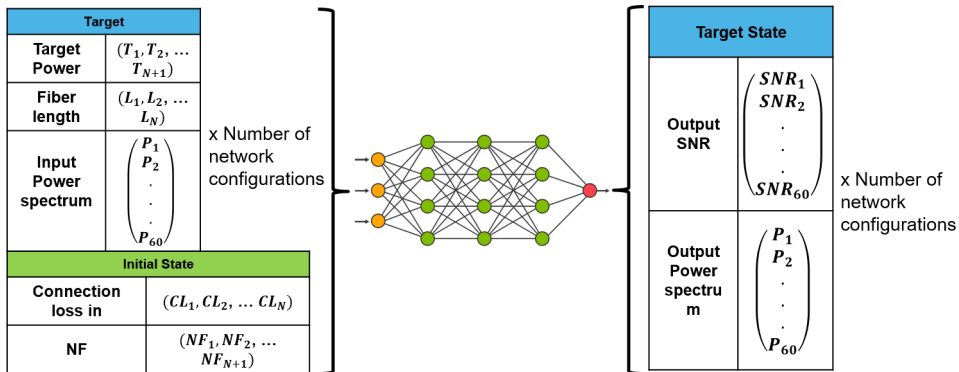


Figure 4.14: LP3 simulation 1: Dataset composition used to retrain the NN model

In this simulation, we assume that we do not know which input parameters are causing the uncertainty. We retrain the model using a data structure as described in Figure 4.14. We generate the dataset by simulating multiple network configurations using the target state parameters, then replace the values of uncertain parameters with the initial state values. This forces the NN model to relearn a completely new mapping between input and output features. We cannot use the initial pre-training dataset in re-training because the behavior of the network has changed and the QoT estimation would not be coherent.

Another factor in the retraining is the number of iterations. If we run the training for too long, we risk over-fitting on the training dataset which would deteriorate the validation error. This trade-off is a question of whether we want to be as accurate as possible on the current network configurations, or generalize for future configurations. For this purpose, we test two methods for determining the number of iterations. In the TRAIN_PRIO method, we prioritize reducing the training error as much as possible and in the VAL_PRIO method we stop the retraining at the point where the validation error is the highest. The training results for each method are presented in Figure 4.15 and 4.16 respectively. For each method, we train the model with an increasingly larger percentage of the training dataset (i.e, increase the percentage by 10% in each step), then use the remaining data for validation. We first retrain the model using only one network configuration to simulate the case where additional data cannot be collected from the network. We generate the training dataset using a single OMS from the FLAT_NET topology. The details of this simulation are described as follows:

- Network topology: OMS AB
- Uncertain parameters: 5 amplifier NFs and 5 fiber CONN_IN
- Number of network configurations: 100 samples
- Batch size: 10 samples from the training dataset
- Loss function: $\text{RMSE}(\text{SNR}) + \text{RMSE}(\text{POWER})$ / for each network configuration in the dataset)

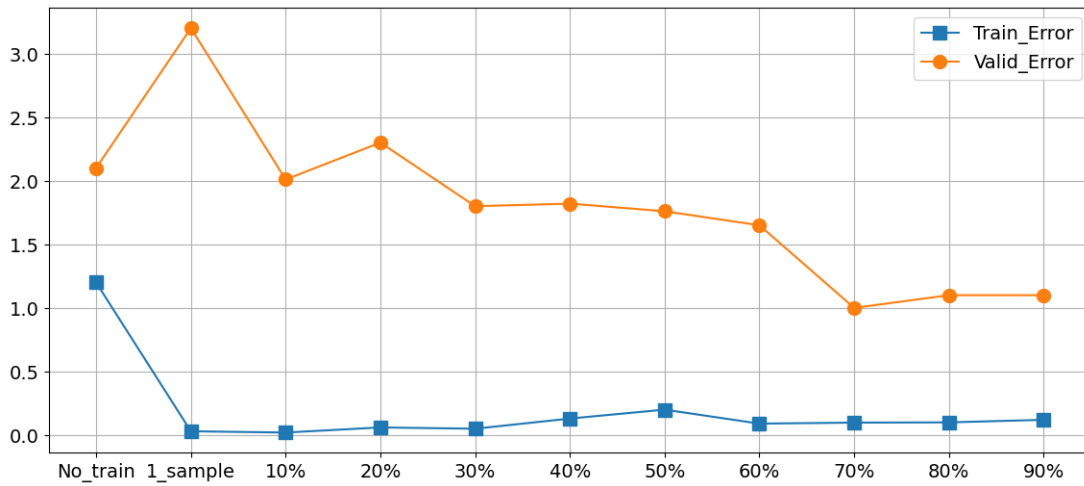


Figure 4.15: LP3 simulation results 1: training and validation results with TRAIN_PRIO method

Using the TRAIN_PRIO method, we observe in Figure 4.15 that the training error is almost less than 0.1 dB compared to an initial error of 1.27 dB. However, the validation error starts at above 2.12 dB, then increases to 3.2 dB in the case of only 1 network configuration, and finally decreases down to around 1 dB. Both the training and validation loss seem to stabilize at 0.1 dB and 1.1 dB when the percentage of training data is superior to 60% and 70% respectively. This seems to be the best performance that we can achieve using this method. However, a validation loss of 1.1 dB is substantial meaning that the model is over-fitting on the training data and cannot generalize well to different configurations.

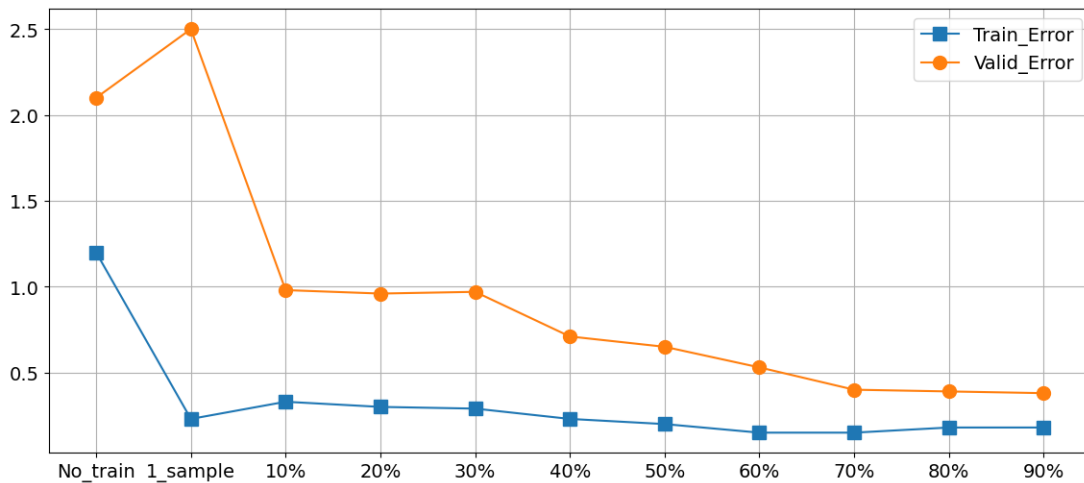


Figure 4.16: LP3 simulation results 1: training and validation results with VAL_PRIO method

Using the VAL_PPIO, we obtain a better match between the training and the validation error. The training error converges to 0.18 dB and the validation error to 0.4 dB from a training percentage of 70%. This latter performance is much better than that using the TRAIN_PPIO method at the same data percentage. Further training after 70% decreases the validation error by 0.6 dB, with only a small (0.1 dB) improvement in the training loss. Furthermore, with only 10% of training samples, the training error is 0.33 dB and the validation error is below 1 dB, meaning we can have a decrease in SNR error of 0.9 dB and 1.2 dB respectively using retraining with a small dataset.

To conclude, retraining the model with no prior knowledge of uncertain parameters does not give satisfying results. This is due to the fact that when the model is retrained, the previous behavior that was learned during pre-training is lost, and the new data is not sufficient to retrain the new behaviour correctly. We did not explore retraining with a dataset size larger than 100 samples because we assume that collecting such dataset is a difficult task in an operational network.

4.5.1.2 Simulation Scenario 2: Retraining the Model With Knowledge of Uncertain Parameters

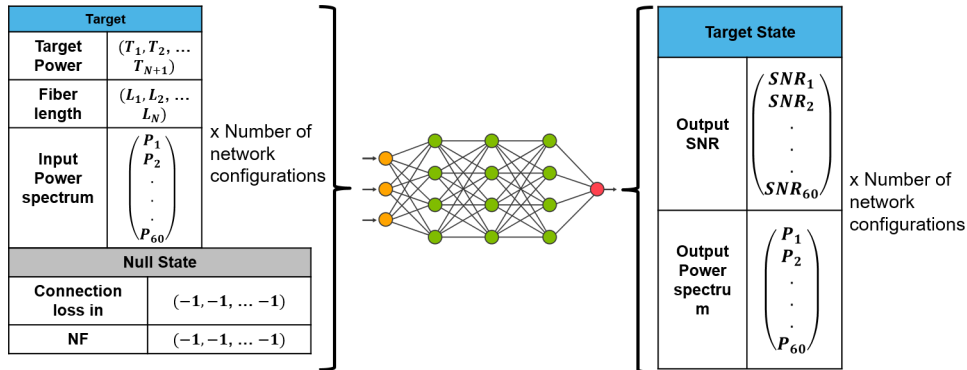


Figure 4.17: LP3 simulation results 2: Dataset composition used to retrain the NN model

If we assume that a certain number of parameters are responsible for the inaccuracy of the QoT estimation, we can use a different training technique than that used in the previous simulation scenario. In Figure 4.17, we show the input and output features of the training dataset in this case. The difference is that instead of setting the uncertain parameter values to the initial state values, we set them to -1 which we call the 'null state'. This state is technically impossible since the NF and CONN_IN parameters cannot have a value of -1 (the NF of EDFA amplifiers is always positive). The purpose of this state is to notify the model that the values of these parameters is unknown. But instead of trying to find their correct values as in LP1, the model will simply learn the corresponding QoT estimation from

the certain parameters. The impact of this change is that we can now retrain the model with data from the pre-training dataset (i.e., the dataset used for the first training having no uncertain parameters) as well as the retraining dataset (i.e., all the datasets having uncertain parameters). The data will contain samples with target states and null states. The model will learn the new behavior of the network through the sample with null states, and it will maintain the previous behavior learned in pre-training using samples with target states. This will serve to regularize the training and force the model to not completely change its behaviour. We test this technique using the following simulation configuration:

- Network topology: OMS AB
- Uncertain parameters: 5 amplifier NFs and 5 fiber CONN_IN
- Number of network configurations: 50 samples
- Batch size: 5 samples from the pre-training dataset + 5 samples from the retraining dataset
- Loss function: $\text{RMSE}(\text{SNR}) + \text{RMSE}(\text{POWER})$ / for each network configuration in the dataset)

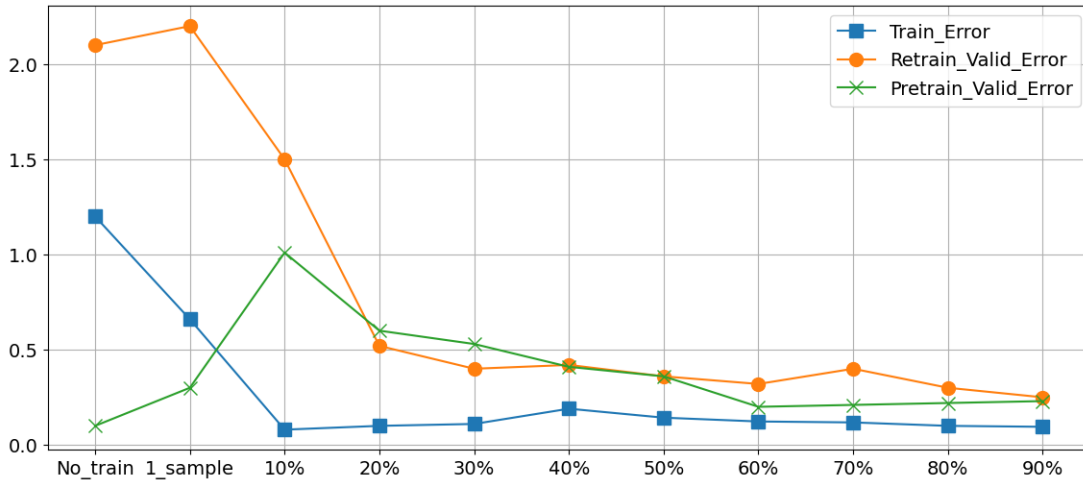


Figure 4.18: LP3 simulation results 2: training and validation results with VAL_PRIO method

We show in Figure 4.18 the results of the simulation. We use the VAL_PRIO method as we want to focus on improving the validation error. We show in the Figure the training

error (TRA_ERR), the validation error on samples from the pre-training dataset 'pre-training_valid_error' (PRE_ERR) and the validation error on samples from the retraining dataset 'retraining_valid_error' (RET_ERR). Before starting the training, only the PRE_ERR is below 0.1 dB as the model was trained on this dataset. Then, after training on a single sample the TRA_ERR is not minimized as observed in the previous simulation but remains at 0.66 dB. This is due to the presence of samples from the pre-training dataset that act as a regularization technique to avoid over-fitting. At the percentage of 10% of training data, the TRA_ERR further decreases down to 0.08 dB, while the PRE_ERR increases to 1 dB. At 20%, both PRE_ERR and RET_ERR are around 0.5 dB while the TRA_ERR is at 0.1 dB. This step is considered as the best compromise between the estimation error improvement and the data set size, because it reaches low error for each estimation error with only 20% of training data (i.e., 10 network configuration samples). To obtain validation errors down to 0.2 dB, we need to increase the training data size to 90% (i.e., almost the full size of the retraining data set equal to 50 samples). Overall the results of this simulation are much better than the previous one. This is attributed to the fact that the NN model's behavior is extended with the new retraining data instead of completely over-write the pre-training behaviour.

4.5.1.3 Discussion Around LP3

The LP3 is based on retraining the NN model to adapt to changes in parameters values. We proposed two methods for building the retraining dataset based on the knowledge of which parameters have changed. The second method turns out to be the most efficient, requiring only 10 new data samples to reach a validation error of 0.5 dB. The retraining does not optimize the uncertain parameters as in LP2 but only assumes that a set of parameters are uncertain. The data requirements can be an issue of this LP and depends of the operational network. If enough margins are available, we can change the configuration of the network manually to collect multiple samples. Alternatively, we can collect data permanently during the life-cycle of the network and detect the network state changes when new services are deployed. However, we risk that the network parameter change significantly during the period of data collection, which would mean that the samples collected are not coherent. In both cases, data collection remains a serious bottleneck for this approach.

The two main factors that impact TL techniques are the generalization of the model and the amount of data required to retrain. Concerning the generalization issue, the model is adapted to a specific case of parameter uncertainty. This means that if the values of the parameters change, the model is no longer usable and needs to be adapted. Furthermore, if we train the model on data from a specific OMS, the model cannot be used on a different one. This means that we have to train a specific model for each OMS. Additionally, the data

required to train the model are of utmost importance due to the lack of data in operational networks as explained in 2.4.

4.6 Third Approach: Detecting Parameter Changes

Unlike the two previous approaches which can be used offline at any time during the life-cycle of the network, the third approach is to detect and correct parameter changes in real time. We first assume that we have perfect knowledge of the network state at the beginning of life of the network. Then, we monitor the QoT measurements in real time in order to detect any changes. Whenever an event occurs and affects the QoT measurements, a learning process is launched to analyse the data, detect changes and identify which equipment or event is responsible for the change. This change can either be diagnosed and fixed by the operational team if it is due to an equipment fault, or it can be considered as a normal change and the corresponding parameter value is updated in the network state.

We assume the following scenario for the learning process. We study a network topology consisting of a single OMS and we assume that the configuration of this OMS does not change in our study. This OMS is vulnerable to two type of events that can alter its performance. For the first event, we assume that one of the amplifiers of the OMS can malfunction, causing the increase of the noise contribution. We model this event by an increase in the NF of the amplifier. In order to simplify the study, we assume that the NF always increases by 1.5 dB. We call this event *Amplifier Deterioration* (AD_{*i*}), the *i* refers to the amplifier that caused the event. The second event is a fiber damage, modeled by an increase in the total fiber span loss. We assume that this increase is equivalent to adding 1 dB to the CONN_OUT parameter of the fiber. We call this event *Fiber Damage* (FD_{*i*}), the *i* refers to the fiber that caused the event. We assume that no other events can cause changes to the network performance. We also assume that a feedback concerning the types of certain number of events (NB_EVE) is given by operator once an equipment diagnosis is performed. This knowledge is mandatory in order to properly train the LP, but it is not required once the model is fully trained. The list of events in the feedback is called 'event chain'. The goal of the learning process is to correctly guess which events caused the succession of performance changes.

In order to perform this detection, we need a ML model that takes as input the configuration of the network (certain parameters), the network state (uncertain parameters) and the QoT measurements before and after the event, then outputs a classification of which event caused the change. After each classification, the network state is updated to reflect the event that was chosen, the model must perform a number of event detection equal to the FD_EV_FDB before it can be corrected. The model must be trained using a feedback

that gives the correct classifications. We have two types of feedback. First, once an event classification is done, we change the network state parameters to reflect the event impact, then we estimate the QoT using a QoT estimation model. The change in the estimated QoT between step i and step $i + 1$ must match the change in QoT measurement that triggered the classification. If it is the case, this means that the new network state is the correct solution or at least an alternative solution. The second feedback is after the diagnostic. The diagnostic identifies the correct event succession, so we can use this information to train the model to better chose the right events instead of simply reducing the QoT estimation error.

We chose Reinforcement Learning models (RL) to implement this approach. Indeed, RL [89] algorithms are able to learn to perform a sequence of actions in an uncertain, potentially complex environment. RL can be applied to any process as long as it can be modeled as a Markov chain. Thus, we use a RL algorithm to sequentially chose the correct event after a set of performance changes. Building a RL model requires the definition of 4 elements: i) the space of actions, ii) the variables that define every possible state of the environment called observation iii) the reward given to the model, and iv) the RL algorithm that learns the policy. The RL model that we use in the simulations of this section is based on the structure in Figure 4.19.

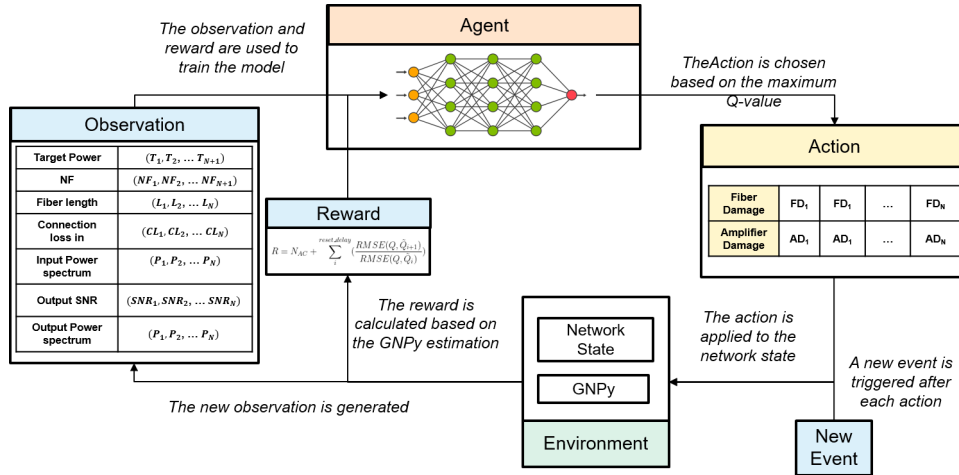


Figure 4.19: Structure of the reinforcement learning model using in LP4

The observation consists of a concatenation of the network configuration, the network state and the QoT indicators before and after the event. We chose the optical parameters similarly to the ones used to train our NN model. The possible actions is a list of all the possible AD and FD events in the network topology. Finally, the reward is based on two parts. First, after each classification. The RL agent is based on a deep Q-Learning model

[90]. It consists of a three-layer neural network model with ReLu activation functions. It takes as input a concatenation of the observations at a step i , then outputs the Q-value of each possible action to take as step $i + 1$. The Q-value is a measure of the potential reward of said action, so in each step, we chose the action with the best Q value. The reward is calculated by accumulating two types of rewards during the training: i)the event reward R_e , which is calculated based on the QoT estimation feedback, and ii) the reset reward R_r that gets added at the end of each event chain. The full reward of an event chain is given by equation 4.1, where N_{AC} is the number of correct classifications in the event chain, and Q and \tilde{Q} are the measured and estimated QoT indicators.

$$R = N_{AC} + \sum_i^{FD_EV_FDB} \left(\frac{RMSE(Q, \tilde{Q}_{i+1})}{RMSE(Q, \tilde{Q}_i)} \right) \quad (4.1)$$

4.6.1 LP4 - Detecting parameter changes using reinforcement learning

We propose a learning process based on the reinforcement learning model presented above. RL models are either trained online or offline. In online training, the model interacts with the live environment and takes actions in real time. This is not an option in the case of optical networks as this would affect the performance of deployed services. In this learning process, we train the RL model to detect the parameter changes using a prepared dataset, then deploy it in the network so it can classify parameter change events. The diagnostic feedback is only necessary in the training phase. Once the model is trained, it can continuously classify events without any reward feedback. Once the model is trained, we can test its performance by applying it to a list of events. In order to simulate this learning process, we set up the following steps:

1. **Event sampling:** We randomly generate 50 different event chains of size FD_EV_FDB . 40 of these chains will be using for training, and the remaining 10 for validating the model.
2. **Model training:** We train the model in multiple episodes. In each episode the model takes actions after FD_EV_FDB events. The observations and rewards for each action is saved as a sample in the training dataset. Once a number of episodes are accumulated ($batch_size$) we run one training iteration. In each training iteration, we split the dataset into batches of size $batch_size$, then train the model using these batches. For example, in iteration i , a dataset of size $S = FD_EV_FDB * (batch_size) * i$ available is used for training, and the model's weights will be updated $S/(batch_size)$ times.

3. **Model evaluation:** We evaluate the model's performance during the training using two metrics:
 - (a) **Event classification accuracy (EV_ACC):** In each training iteration, we calculate the number of correct classifications divided by the number of total training samples
 - (b) **QoT estimation error (QoT_ERR):** After each episode during the training process, we calculate the SNR estimation error after all the actions we performed. This metric shows if the model is learning to reduce the error on SNR estimation regardless of the accuracy on the event classification. We chose to calculate this error at the end of each episode instead of after each action, because the estimation error can be accumulated by a succession of wrong classifications.
4. **Model validation:** After the model is trained, we validate it's performance by testing it on an event chain of 100 events. We calculate the EV_ACC and QoT_ERR metrics over this event chain.

4.6.1.1 Simulation Scenario 1: Classification with 3 possible events

We first tested the model's performance in the smallest possible network scenario. We run the simulation on OMS AB of the FLAT_NET topology, but consider only three types of events; FD in the first span, AD in the booster and AD in the preamp called [FD1, AD1, AD2] respectively. We run the simulation with the following configuration.

- Network topology: OMS AB
- Possible events: [FD1, AD1, AD2]
- FD_EV_FDB: 10
- Batch size: 20

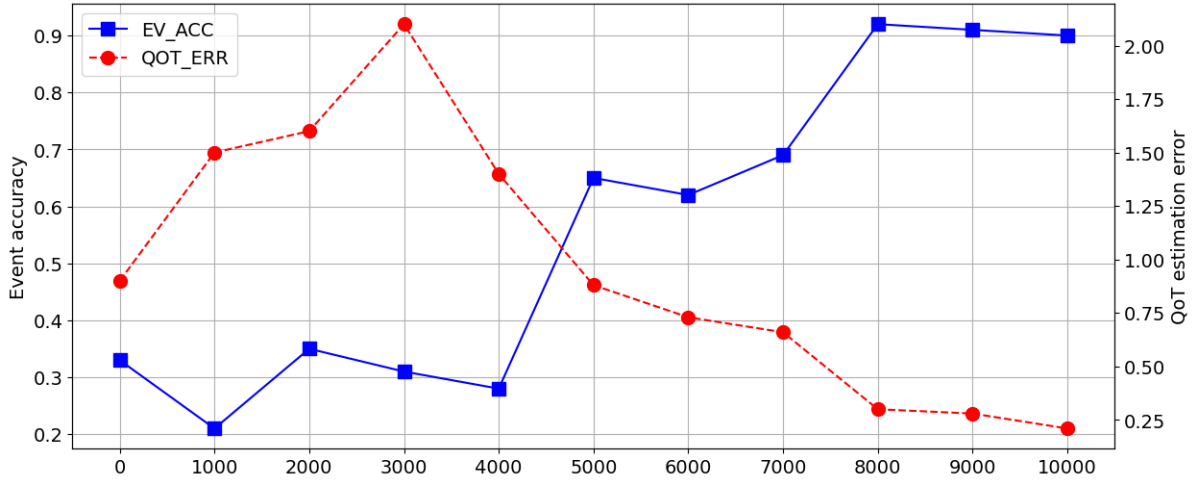


Figure 4.20: LP4 Simulation 1 results: Evolution of the EV_ACC and QoT_ERR at different milestones in the training phase

In Figure 4.20 we show the training results after each thousand iterations. At each milestone k , we report the evaluation metrics of the $(k * 1000)^{th}$ iteration. The first step is the evaluation right before the first training iteration. The accuracy starts 33% then and remains below this value for the first 4 milestones. The accuracy then jumps to 66% at the 5th milestone. It remains below 70% for the two next steps, then jumps again above 90%. In comparison, the QoT error decreases slowly after the 3rd milestone until it reaches 0.21 dB. We observe that it takes up to 9000 iterations for the model to start performing correctly. In the end of the training both metrics do not continue improving, so we did not push the training further than 10k iterations. In general we can say that the model learned to both minimize the QoT error and classify the events correctly.

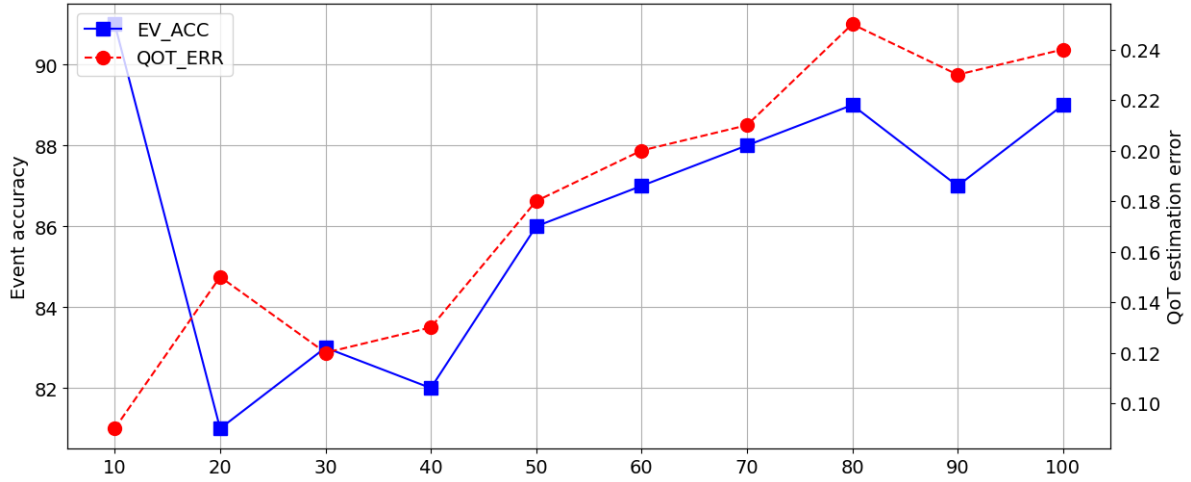


Figure 4.21: LP4 Simulation 1 results: Validation results

In Figure 4.21 we show the results of the validation metrics for our simulation. We show the evolution of the SNR estimation error, alongside the event classification accuracy. We notice that the SNR estimation error starts at 0.09 dB, but slowly increases up to 0.25 dB. The classification accuracy fluctuates slightly in the beginning, then stabilizes at 89%. This means that the model has a stable accuracy throughout the validation, but the mistakes in classification accumulate which slowly increases the QoT estimation inaccuracy. If we can have regular diagnostic feedback, we can reset the network state after a number of events so that the SNR estimation error never reaches high values.

4.6.2 Discussion Around LP4

In this section, we proposed an LP that can detect changes in parameters in real time. The model learns to reduce the QoT error and correctly classify the source of performance change using a reinforcement learning process. The classification accuracy of the model decreases when the number of events increases. However, even if the event type is not well chosen, the model is still able to reduce the QoT estimation error. Additionally, the RL model needs a substantial amount of data to be trained on. It is not possible to collect this data in real field as it is hard to isolate events in a real scenario. We can simulate the event in a controlled environment in the lab, or simulate the events using a QoT model as shown in this section.

4.7 Conclusion

In this chapter we proposed three approaches to solve the parameter uncertainty issue in QoT estimation. Each approach has its advantage and its drawbacks. The first approach uses optimization algorithms to refine the values of the uncertain parameters in order to minimize the QoT estimation error. This approach does not require any additional data, other than the initial QoT measurements. We have shown that it can be applicable to both an analytical and machine learning based QoT models. The learning processes implementing this approach can minimize the QoT estimation error, as long as the parameters being optimized contribute to the impairment which is the source of the error. The drawback of this solution is its inability to manage a high number of uncertain parameters. In fact, in this case the solution could minimize the QoT error but could be unable to retrieve the accurate values of parameters. To overcome this issue, the LP1/LP2 could be performed periodically during the life-cycle of the network in such manner that the algorithm detects faster the change in uncertain parameters and the refined values become the initial values during the next execution of the learning process.

The second approach relies on retraining a ML-based model in order to be adapted to measurements changes. In this case, the model is less dependent on the choice of uncertain parameters and can learn new behavior by modifying its weights. However, the retraining requires a large amount of data to be collected, which might not be possible in an operational context.

The final approach is based on detecting parameters changes through failure event classification. We have shown that using a RL algorithm we can train the model to detect which equipment have changed and keep the QoT estimation error low. The main drawback of this approach is that it is hard to scale to larger network topologies, and it requires precise knowledge of the different failure events and diagnostics in order to be trained properly.

The choice of approach depends on the operational needs and constraints. The first approach is efficient in the case where the QoT model is robust and the number of uncertain parameters is limited. The second approach can be interesting in the case where many QoT measurements are available to train the model. The third approach is more adapted if the goal is to perform soft failure detection as well as a QoT estimation improvement.

Field Trial to Assess Bayesian Optimisation Based Approach

Contents

5.1	Introduction	100
5.2	Data Extraction	100
5.2.1	Extraction Tool	100
5.2.2	Data Description	102
5.2.3	Experiment Sub-Network Topology	104
5.3	Experiment Results	105
5.4	Conclusion	108

5.1 Introduction

In the previous chapter, we presented four learning processes to reduce the QoT estimation error due to parameters uncertainty. We have shown the performance of these LPs by testing them in different simulation scenarios using the GNPpy tool. LP1 seems the most mature and the easiest to apply in a live network compared to the others since it does not require an NN-based QoT estimator as LP2 and LP3, or the event feedback information necessary to train the RL model in LP4. To further validate its performance, we test this LP in a live network with real measurements. Data are collected using a dedicated tool that connects to the NMS of the operational optical transport network and collects data related to the topology, services and performance. We applied the BO-based parameters optimization of chapter 4 on a sub-network topology. We present in section 5.2 the data extractor tool, the collected data and the network topology we used in our experiment. In section 5.3, we show the results of the experiment that we ran on the extracted data. Section 5.4 concludes this chapter. The field trial results were published in a conference paper [16].

5.2 Data Extraction

5.2.1 Extraction Tool

To run the the learning processes proposed in chapter 4, we need QoT measurements and parameter for the network state. To extract this data, it is possible to use either the North-bound Interface (NBI) from the NMS which controls all the equipment, or directly from the equipment through the Management Interface of the network element. Extraction based on the Management Interface is not mature yet and protocols like gRPC or NETCONF are not yet adopted by all vendors. Access to equipment data is possible via certain legacy protocols such as SNMP which are generally used for test and validation purposes but are not suitable for production and large-scale network context.

On the other hand, the NBI-based extraction solution is currently achievable through SOAP and RESTful APIs provided by vendors. The NBI offers to an external application the possibility to interrogate the NMS via a RESTful HTTP service. To extract a large amount of data, the use of the NBI is the most convenient way for the time being. For this reason, a Data Extraction tool have been developed in our team to exchange with the NMS and extract data related to the network topology (e.g., nodes, cards and links), the performance in each port/device (e.g., emitted power, bit error rate at the receiver) and the alarms. The Data Extractor tool was connected to an operational network and real data were collected. We have used this data in the following experiment presented in this

chapter.

The Data Extractor tool involves several extraction modules. Each is targeted to the NMS of a different vendor, because the data model and API exposed is specific to each vendor. In order to collect the needed data, the Data Extractor launches a set of jobs. Each job is intended to collect data related to specific target among these four: topology, service, performance or alarm. In our study, we do not consider data related to alarms. In each job, several requests are sent to the NMS using the REST protocol. Some requests do not support REST, so the SOAP protocol is used instead. The NMS server processes the request sent by the Data Extractor and returns a response in an expected JavaScript Object Notation (JSON) format. The Data Extractor filters out the relevant data and stores them in CSV files. The obtained CSV files are timestamped and well-structured, which facilitates their use by a ML-based algorithm. The jobs are launched periodically. The period depends on the change frequency of the targeted elements in the network. For a real network, the job configuration is done as follows: topology/service jobs are launched each week, whereas performance and alarms jobs are launched each 15 minutes.

The NBI APIs are based on proprietary data models which require the implementation of a data extraction module per vendor and additional module that converts the extracted data to the target data model. In our case, we need to convert the extracted data related to the topology and services to the GNPY data model. Therefore, we use a software converter that generates a GNPY compliant input files from the obtained data.

Data Extractor is a useful tool to obtain data but it is worth highlighting these following points that give some feed-backs concerning the data acquisition in a real network:

- The data extraction process must guarantee speed, lightness and confidentiality.
- A substantial effort is needed to develop a data extractor taking into account the specificity of each vendor data model and API.
- The proprietary data models requires the implementation of a data converter module per vendor to adapt the extracted data to the target data model. Moving from a proprietary API towards a standardized one common for all vendors will simplify the implementation of these kind of tools and optimize the development and test efforts.
- NBI API is not suitable to collect huge amount of data (networks with more than 1000 nodes) in real time and a dedicated performance monitoring protocol should be proposed in order to efficiently extract real-time data from equipment.
- The data provided by the NBI suffer from outlier and missing values although most of these missing data are available in the NMS graphical user interface. This means that the NBI is not perfectly synchronized with the NMS databases.

5.2.2 Data Description

Table 5.1: List of features extracted from the NMS

Target	Items	Example of features
Topology	List of Fibers	A-Port ; Z-Port ; Span Loss ; Fiber Type
	List of nodes	GUI label ; site name; position ; node type
	List of NE	GUI label ;NE id ;NE name ; NE type ; NE model
	List of cards	card name ; card Type ; slot location ; NE ID ; NE Name
Service	List of paths	GUI label ; order id ;layer rate ;container type ;operational state ; A-port ; Z-port ; A-NE ; Z-NE
	List of OS	GUI label ; layer rate ; A-port ; Z-port ; A-NE ; Z-NE
	List of OUT	GUI label ; layer rate ; container type ; A-port ; Z-port ; A-NE ; Z-NE
	List of ODU	GUI label ; layer rate ; container type ; A-port ; Z-port ; A-NE ; Z-NE
	List of OMS	GUI label ; layer rate ; container type ; A-port ; Z-port ; A-NE ; Z-NE
	List of OTS	GUI label ; layer rate ; container type ; A-port ; Z-port ; A-NE ; Z-NE
Performance	Common fields	Port Name ; Port Label ; Port Rate ; Port Type ; NE Id ; NE name ; Card name ; Card Id ; Card type ; Shelf type ; Connected from ; Connected to
	Transponder	GUI label ; layer rate ; PostFEC BER ; PreFEC BER ; Tx-Rx power ;
	Amplifier	input power ; output power ; gain ; ...

Despite the challenges outlined above, we managed to extract data from a live network consisting on more than 1000 nodes using the extractor. Unfortunately, due to confidentiality and missing data issues, not all the extracted data was exploitable. We show in Table 5.1 the list of features that we were able to extract. The features are split into topology, service and performance features and were processed and converted into GNPY data format in order to prepare them for simulations. As discussed in section 3.5, we consider a network state based on GNPY input parameters. When analysing the data, we observed some missing or incomplete parameters. This was either because we were not able to collect them from the NMS using the API or simply because they were not measured contrary to what we assumed before (e.g., BER of some cards). In Table ??, we propose a similar parameters Table to that we presented in section 3.5 but we update it to reflect the availability and certainty of the extracted parameters.

Table 5.2: Network state from extracted parameters

	Parameter/QoT indicator	GNPy	Data Extraction
TRx	Service configuration (Bit rate, Baud Rate, Spacing ...)	Individual service parameters	The Bit rate, Baud Rate, Spacing and frequency are available
	Back-to-back penalty	OSNR penalty	Not available in the controller. We used characterization data provided by equipment vendors
	Bit Error Rate (BER)	Not Computed	The BER is only available for a limited list of TRx boards
	Signal to noise ratio (SNR)	Computed	Certain: deduced from the BER
ROADM	Equalization	Fixed PCH or PSD	No equalization information was available. We assumed that a PSD-based equalization is performed, but we consider the PSD value unknown
	WSS configuration (signal loss, filtering penalties ...)	Not required	Not extracted
EDFA	Noise Figure	Required	We consider it uncertain, and use the default value from the equipment sheet as an initial value
	Total Gain	Required	The total gain value can be extracted. No information about the gain spectrum is available
	Total input power	Required	Can be extracted
	Total output power	Required	Can be extracted
	Per channel input/output power	Required	No power spectrum is available in any amplifier
Fiber	Total loss	Required	The total loss was deduced from the total power difference
	Loss distribution	Required	Fuses or connection losses is available in other database
	Fiber coefficient (attenuation, dispersion, gamma, PDL)	Required	Assumed the default values per fiber type

Parameters related to transponders were always available except for the BER measurements as only a small number of cards were reporting the BER through the API. We were not able to extract any parameter about ROADMs, because the ROADM component was considered as passive and no data was available via the API. Total power measurements were available at the input and the output of each amplifier, but no power spectrum was available. This means that we could calculate the total span loss, so we used the `CONN_IN` parameter to model the fiber distribution. However, we were not able to measure the power spectrum at the ends of the OMS. We assume that the WSS equalization was performed using a PSD parameters based on the spacing of each channel, but we did not have access to the value of the PSD. For this reason, in addition to `NF` and `CONN_IN` parameters, we consider the value of the equalization PSD as uncertain. This new parameter would allow us to have an approximation of the WSS output power spectrum in order to launch the simulations using GNP_y, and replace the second step of the optimization in LP1 and LP2 (spectrum power shift). We use this network state configuration in the following experiments.

5.2.3 Experiment Sub-Network Topology

After extracting data, we limited our experiment to a sub-network consisting of two optical transmission lines (`EXTRACT_NET`). The network is managed by an NMS, composed of 8 ROADMs and extended over 1000 km. These two lines are independent of each other and have different equipment configurations. Each OMS is composed of fiber spans of different lengths as depicted in Figure 5.1.

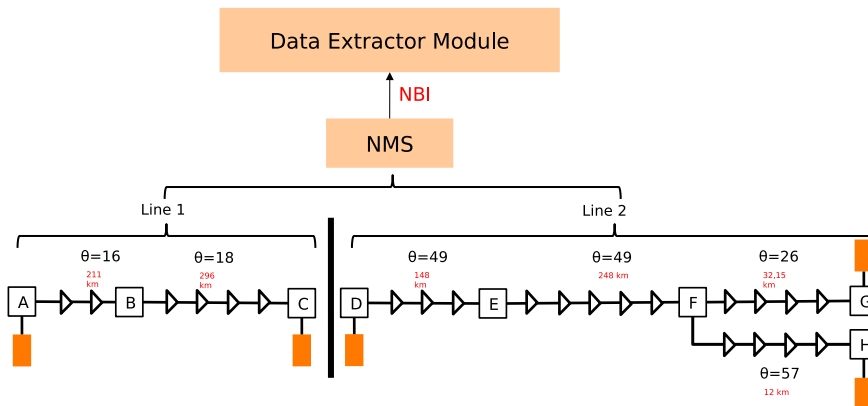


Figure 5.1: The sub-network transmission lines used for the optimization process

Data Extractor tool is used to collect data via the NBI of the NMS to provide data on the topology, the deployed services, the equipment configuration and the performance metrics. Data extracted from the NMS are not exhaustive, which leads us to complete the models with external information such as equipment characterizations. The performance data provide the output power of each amplifier and the BER of each service. The BER is transformed to SNR using transponder back-to-back characterization. Several services (between 16 and 49) are transmitted through the setup with different source-destination nodes leading to a different spectrum load distribution in each OMS. For this experiment, we only monitor the performance of a selected list of services, 12 services in Line 1 and 8 services in Line 2, as depicted in Tab. 5.3.

Table 5.3: List of monitored services in the two transmission lines

Line	Route	List of Services
1	A-B	S_{11}
	B-C	$S_2, S_3, S_4, S_5, S_6, S_7, S_8, S_9$
	A-C	S_1, S_{10}, S_{12}
2	D-F	S_1, S_7, S_8
	D-G	S_2, S_3
	D-H	S_4, S_5, S_6

5.3 Experiment Results

For this experiment, we use the learning process LP1 based on Bayesian Optimization to optimize network parameters. For this purpose, we set up a service deployment scenario in order to emulate how the LP can reduce the SNR estimation error during the life cycle of the network. Since the monitored services already exist in the field, we assume that all the services are already deployed, but we progressively increase the number of available measurements for the LP.

We apply the model as described in 4.4.1. We define S_D as the set of services already deployed and S_{ND} as the set of services not yet deployed. In each step, we perform the optimization based on performance collected from S_D , then we compute the training error and the validation error corresponding to the error between measured and estimated SNR in S_D and also in the services planned to be deployed (i.e., S_{ND}), respectively. This method allows us to test the performance of the LP with a different number of QoT measurements, and to calculate what is the gain in QoT estimation error right before deploying a service.

Graphs in Figure 5.2 and Figure 5.4 depict the progression of the mean SNR error after each successive optimization for Line 1 and Line 2 respectively. The initial error (int-err)

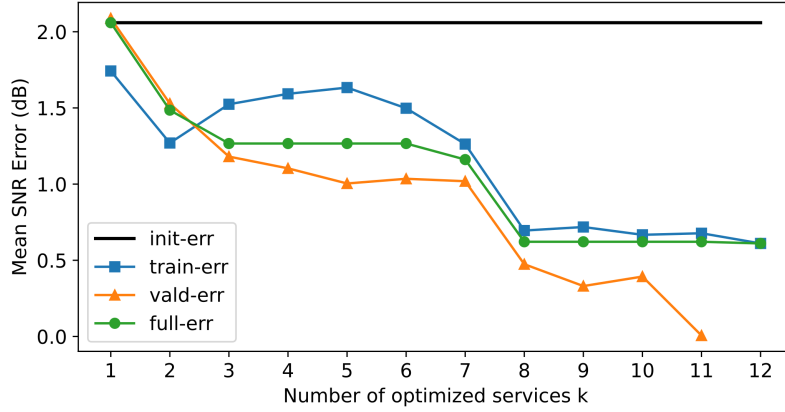


Figure 5.2: Progressive deployment scenario - Mean SNR error progression for Line 1.

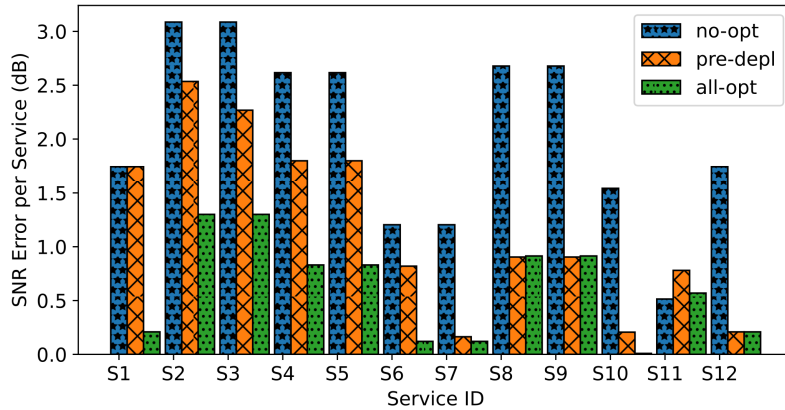


Figure 5.3: Progressive deployment scenario - SNR error per service for Line 1.

shows the expected performance without the optimization process and serves as benchmark. The training error (train-err) and validation error (vald-err) are calculated at each step k by averaging the SNR error for S_D and S_{ND} services respectively. We notice that the initial error is around 2 dB for Line 1 and 1.5 dB for Line 2. Those errors are mainly due to the fact that data extracted from the live network and used to model the two lines in GNPY are not reliable (e.g., missing or not up-to-date values). The high initial error stresses the importance of refining network parameters. After performing the optimization, the mean SNR error (i.e., full-err) is reduced by 1.44 dB for Line 1 and 0.52 dB for the Line 2. This difference in performance is due to the initial error, the number of parameters to optimize and the number of already deployed services in each case. The training and validation error

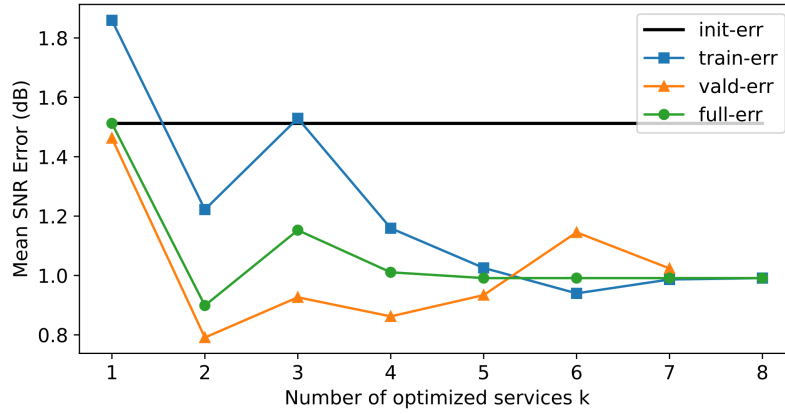


Figure 5.4: Progressive deployment scenario - Mean SNR error progression for Line 2.

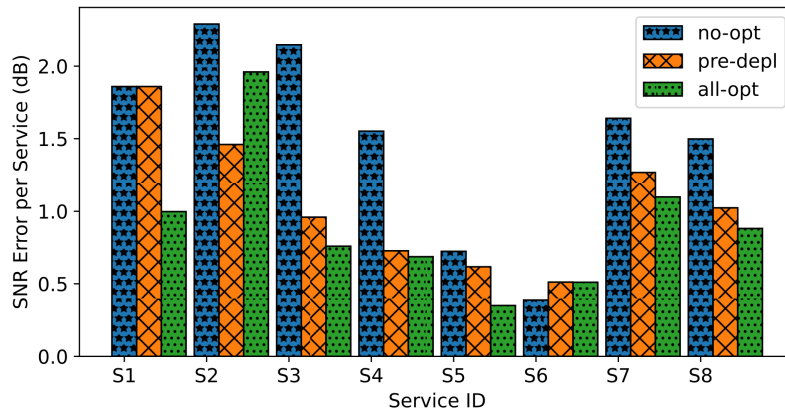


Figure 5.5: Progressive deployment scenario - SNR error per service for Line 2.

slightly fluctuate depending on which service is optimized first, but the overall trend shows that the error decreases proportionally to the number of services taken into consideration during the optimization.

In order to deeply understand the optimization result, we plot in the graphs of Figure 5.3 and Figure 5.5 the improvement in the SNR estimation for each service. The error without optimization (no-opt) is the initial error per service. The error before service deployment (pre-depl) shows the expected improvement in the SNR estimation if the optimization process is performed just before deployment. The error after all optimizations (all-opt) is the expected performance after all optimizations are performed. We notice that the initial SNR error for each service varies between 0.5 dB and 3 dB. After successive optimizations, the

performance of services having an initial error over 0.5 dB is significantly reduced and the improvement in SNR error can reach up to 1.78 dB. Furthermore, the all-optimization error shows that continuous optimizations serve to increase the accuracy of the QoT estimation further and does not penalize already deployed services. By analyzing the pre-depl error, we can expect up to 1.6 dB improvement in QoT estimation of services right before deployment.

5.4 Conclusion

In this chapter, we tested the LP1 based on the Bayesian optimization to refine uncertain parameters in a live network. Therefore, we used a data extraction tool that collects data directly from the controller. This proves that our approach can be successfully applied in the operational network without further telemetry. We were not able to test LP2 and LP3 because they are based on a neural network model that should be trained using different data distributions than the collected ones. LP4 was not also tested because we do not have data related to failure event feedback that is needed to train the RL model. Using LP1, we show that the accuracy of QoT estimation is improved by up to 1.78 dB per service. The reduction in QoT estimation error is substantial, but the results could be improved by adopting streaming telemetry for data extraction.

Chapter **6**

Conclusion and Future Work

Contents

6.1	Conclusion	110
6.2	Perspectives	114

6.1 Conclusion

Optical transport technology is being pushed to work closer to its theoretical limits to satisfy the ever-increasing traffic demands. The increasing complexity of these systems will make their design, control, and management challenging. On the other hand, a large amount of data is currently available in industrial Network Management Systems (NMSs), but it is not well harnessed by operators. The goal of this thesis was to explore ML techniques in order to open new opportunities to utilize these data to optimize optical transport networks and enhance their reliability. We specifically focused on the QoT estimation topic, since an accurate QoT estimation is required for various tasks related to the management of optical networks such as service provisioning and network design. QoT estimation is performed using QoT models that rely on a set of network parameters to predict the quality of the optical signal. An accurate QoT model and a reliable network state representation allow the operator to estimate the potential impact of any configuration changes before deployment, which could act as a module of the digital twin for the transport layer.

We carry out the study from an operator point of view. In other words, we aim to propose solutions that can easily be deployed in the operational network, using only data in the management/control plan. This does not preclude that, in some parts of this study, some assumptions related to futuristic data extraction methods (e.g., data monitoring) are considered.

We began our study with a literature overview, where we surveyed multiple papers that attempted to use ML to improve the QoT estimation. We found that ML models are used in a variety of ways to train QoT estimators, enhance analytical models and deal with data shortage issues. We discussed the validity of their approaches and assumptions when it comes to a real operational scenario. The main issue that we observed is the high data requirements. Indeed, in order to train some proposed models, a substantial monitoring and a data collection infrastructure must be set up. Furthermore, some solutions relying on ML to potentially replace analytical models do not bring sufficient gains to justify the additional cost of their implementation and deployment.

Analytical models are performing very well when their physical assumptions correspond to the reality on the ground and their input parameters are accurately provided. But, in the case when these two conditions are not met, ML models can be used to improve the QoT estimation by either estimating some difficult-to-characterize optical impairments, or by refining the hard-to-measure network parameters. Additionally, a number of papers have focused on transfer learning approaches to reduce the amount of data needed for training. This could be an important factor that makes ML-based model industrialized in the future. As a conclusion of our literature overview, we consider that ML models can be more useful

in the short / medium term if they work alongside analytical models, although we do not rule out that full ML-based QoT estimation could be improved to reach high performance and be easily generalized. For this reason, we have used in our simulations both a ML-based model and an analytical QoT model.

Given our conclusions from the literature study, we have focused on the issue of network parameters inaccuracy. We mean by uncertain parameters, those which are missing or do not have reliable values. This issue affects the accuracy of QoT estimation tools based on either analytical or ML models. To resolve this issue, we benefit from the large amount of QoT measurements existing in the network, particularly BER measurements. Actually, these data have not been used before to improve the accuracy of QoT estimation. Therefore, our approach aims to create a closed-loop process to compute QoT. The *feed-forward* part of the closed-loop is the QoT estimation tool, while the *feedback* part is our ML-based solution that uses BER and SNR measurements on already-deployed-optical services to improve the accuracy of QoT estimation of future services.

The identification of uncertain parameters depends on the ability of equipment to detect and update changes of their parameters and the ability of the control/management plan to collect those parameters. According to the study that we carry out to identify uncertain parameters, we have assumed three main sources of inaccuracy. Firstly, we consider amplifier NF as the first uncertain as it represents the main source of ASE noise. Secondly, we consider the fiber loss distribution as the second uncertain parameter and model it using a combination of connection and fused losses. In fact, the nonlinear effects can be accurately calculated as long as the power level along the length of the fiber is well characterized. While the emission power could be accurately measured, the fiber loss is difficult to estimate due to uncertain lumped loss (i.e., connection loss and fiber splice loss). Finally, we consider the spectrum power shift as the third inaccurate parameter since it impacts the per channel QoT estimation. Using this set of uncertain parameters, we compensate for most of the sources of inaccuracy impacting the QoT estimation.

To resolve the parameters uncertainty, we proposed three approaches. The first approach aims to refine uncertain parameters at the input of QoT estimation tool. In the case where an analytical QoT model is used (GNPy is taken as an example), we propose a learning process based on Bayesian optimization (i.e., LP1). We observed that LP1 can minimize the SNR estimation error to lower than 0.1 dB for all the services that it was trained on. When we apply it to new services in a different network configuration, the error is still reduced by up to 0.8 dB, and the residual SNR error can remain up to 0.23 dB. This means that the LP is accurate on the services already deployed, but can still have a small estimation error in the prediction of QoT for the services to be deployed. In the case where a ML-based QoT estimation is used, we propose a learning process based on a gradient descent algorithm

(i.e., LP2). LP2 is able to reduce the SNR error to less than 0.1 dB with the training configuration, but with the redesign configuration, this SNR error can reach 3 dB. This first approach performed the best and required the least amount of data to be trained. It can reliably be used in the operational network to reduce the QoT estimation error before service deployment. When the network configuration changes, the first few QoT estimations might have some inaccuracies, but once new measurement data is available, the LP can be run again to give better performance.

the novelty in this approach lies in its operator-centric perspective, which sets it apart from existing literature. This perspective brings two key differences. Firstly, we consider uncertain parameters by carefully analyzing operational constraints, allowing for more realistic representations. Secondly, we employ optimization techniques specifically tailored for black-box QoT tools. While our methods demonstrate performance comparable to similar studies, we further validate their effectiveness by applying them to live network data. This additional validation enhances the credibility and applicability of our approach.

The second approach aims to improve the QoT estimation by changing the QoT model itself. We proposed a learning process (i.e., LP3) that retrains a ML-based QoT model to adapt it to inaccurate input parameters. The LP can relearn the new behavior without finding the correct values of the uncertain parameters. Once the model converges, the training error is minimized below 0.1 dB and the validation error below 0.25 dB. The main advantage of this LP is that it can also compensate for any other inaccuracy in the QoT estimation without necessarily knowing where the inaccuracy comes from. In this LP, the weights of the model itself are changed to take into account the inaccuracy without identifying the uncertain parameters. The drawback of LP3 is that the retraining requires multiple new sample in order to work. Even if we compromise in the QoT estimation accuracy, we still need at least 10 new samples. Note that each sample requires a different network configurations from the others. Thus, the applicability of this approach depends mainly on data collection availability.

In the literature, transfer learning (TL) techniques have been utilized to adapt QoT models to various domains, as outlined in Chapter 2. However, our contribution uniquely applies TL to specifically address parameter uncertainty. Unlike other studies that have employed TL techniques to adjust models to changes in parameter distribution (as in [69]), assuming accurate re-training data, our approach differs in its assumptions, leading to distinct TL tasks and making direct performance comparisons challenging. Nevertheless, the surveyed papers indicate that TL can be achieved with even less data than we have used, suggesting potential for enhancing our contribution through more sophisticated TL techniques.

The third approach consists of detecting parameter changes in real time. If we can keep

up with each change in performance then detect which parameter change caused it, we can maintain an accurate estimation of the QoT. This approach could be achieved in the frame of a digital twin where the parameters values are updated as changes occurs, creating then an accurate database of the network state. To implement this approach, we propose a learning process based on a reinforcement learning (LP4) able to classify events that cause parameter changes, based on the variation of the performance. We train this model to detect fiber and amplifier damage, which we represent using variation in the connection loss and NF respectively. LP4 performs well in the case of a small number of events. It can reliably detect the events and reduce the QoT estimation error. However, the classification accuracy decreases when the network topology is large, even though the model can still maintain an estimation error around 0.5 dB. This approach could be used to detect failure, but it requires collecting diagnostic feedback data in order to train Reinforcement Network algorithm on all potential failure events in the network.

This approach is indeed quite unique in the literature. While most studies tend to focus solely on either fault management or QoT estimation improvement, our contribution aims to combine both tasks using a reinforcement learning (RL) agent. It is worth noting that when considering fault classification and QoT improvement as separate metrics, other papers may demonstrate more significant results than our contribution. However, we believe the novelty of our work lies in the correlation we establish between QoT inaccuracy and fault detection, which adds a valuable dimension to this research work.

In addition to simulation experiments, we tried to test these approaches using data collected from a field trial. The available data enabled us to run only LP1 based on Bayesian Optimization. We adapted our learning process by adding an additional uncertain parameter which is the equalization PSD as we do not have any idea about the power management in the output of the WSS. The results of the experiment showed that despite the lack of exhaustive network state, LP1 is able to achieve a gain up to 1.7 dB in SNR estimation error. This experiment was a precious opportunity to challenge our learning process in a real field scenario. This experiment is to the best of our knowledge represents the first time where SNR improvement was reported using data collected from the network management system.

This field trial has provided valuable insights into the deployment of ML-based solutions in a live network. Firstly, despite conducting a thorough analysis of parameter availability, the collected data did not align perfectly with our expectations. This provides evidence that solutions aimed at improving QoT estimation should be capable of adapting to the diverse levels of information available in the network. In Chapter 5, we demonstrated the flexibility of the Bayesian optimization LP by introducing a new uncertain parameter (PSD) and utilizing only SNR as an objective function.

Secondly, we observed a significant margin for QoT improvement in the real network. The initial SNR estimation error exceeded 3 dB, mainly due to substantial inaccuracies in network parameters. This further justifies the need of a such data refinement tool for parameter optimization. It is important to note that the margin of improvement may vary depending on the network condition and data accessibility. Nevertheless, we have demonstrated that even in challenging conditions, our learning process can enhance QoT estimation.

Thirdly, due to limitations in data collection, we were unable to apply several of the LPs proposed in our research. This highlights a significant bottleneck that hinders the deployment of ML-based solutions. Our key takeaway is that without substantial advancements in telemetry and monitoring, the successful implementation of most ML-based solutions from the literature would be challenging in operational networks.

6.2 Perspectives

The learning processes we proposed show promising results in simulation and real network scenarios. However, there is always room for improvement. These following steps summarize the potential perspectives of this thesis:

- **Explore different network scenarios:** Two additional use-cases could be further investigated. The first one related to C+L band [91]. In this use-case, the modeling of the connection losses is different because we have more equipment per band, moreover, the SRS becomes more severe than in the C-band case. The second use-case concerns the integration of Raman amplifiers [92] that will change the power level progression in the fiber so we might need to change the model of the lumped loss distribution.
- **Improve the parameter optimization LP:** The uncertain parameters we used cover a large amount of uncertainty on the QoT estimation. Other parameters could be taken into consideration such as transponder back-to-back uncertainty. Furthermore, our Bayesian optimization approach could be tested with alternative analytical model-based QoT estimation tool other than GNPY. Also, in order to reduce the number of parameters to optimize (e.g., a few dozen of parameters), the LP1/LP2 could be permanently executed jointly with LP4 during the life-cycle of the network. Once LP4 detects a change, LP1/LP2 will be run in order to refine the parameters. In this case, the LP1/LP2 will have few number of parameters to refine.
- **Improve NN retraining:** The third learning process (LP3) can be improved by using different transfer learning techniques. Currently, the LP requires a large amount

of data for retraining, this can potentially be reduced using techniques such as one-shot learning. Furthermore, the NN model used to estimate the QoT is simple. If we can test LP3 with a more efficient QoT model, this might improve the performance of the learning process.

- **Improve the Reinforcement Learning Process:** In the LP4, we proposed a way to detect parameter changes, but we made multiple assumptions related to the failure events, for instance only two type of events are considered. In order to make it applicable in the field, we need to relax some of these assumptions which make the model easily generalizable. However, this requires a more detailed knowledge of the failures occurring in the field and obtaining performance data that correlates with alarms and maintenance data generated by the operational team.

References

- [1] Mayur Channegowda, Reza Nejabati, and Dimitra Simeonidou. “Software-Defined Optical Networks Technology and Infrastructure: Enabling Software-Defined Optical Network Operations [Invited]”. en. In: *Journal of Optical Communications and Networking* 5.10 (Oct. 2013), A274. ISSN: 1943-0620, 1943-0639. DOI: 10.1364/JOCN.5.00A274. URL: <https://www.osapublishing.org/abstract.cfm?URI=jocn-5-10-A274> (visited on 09/29/2021).
- [2] Victor Lopez et al. “Transport API: A Solution for SDN in Carriers Networks”. en. In: (), p. 3.
- [3] Martin Birk et al. “The OpenROADM initiative [Invited]”. en. In: *Journal of Optical Communications and Networking* 12.6 (June 2020), p. C58. ISSN: 1943-0620, 1943-0639. DOI: 10.1364/JOCN.380723. URL: <https://opg.optica.org/abstract.cfm?URI=jocn-12-6-C58> (visited on 01/31/2022).
- [4] Shoichiro Oda et al. “A Learning Living Network With Open ROADMs”. In: *J. Lightwave Technol.* 35.8 (Apr. 2017), pp. 1350–1356. ISSN: 0733-8724, 1558-2213. DOI: 10.1109/JLT.2017.2660540. URL: <http://ieeexplore.ieee.org/document/7835740/> (visited on 01/08/2020).
- [5] M. Dallaglio et al. “Control and Management of Transponders With NETCONF and YANG”. en. In: *Journal of Optical Communications and Networking* 9.3 (Mar. 2017), B43. ISSN: 1943-0620, 1943-0639. DOI: 10.1364/JOCN.9.000B43. URL: <https://opg.optica.org/abstract.cfm?URI=jocn-9-3-B43> (visited on 01/31/2022).
- [6] Francesco Paolucci et al. “Network Telemetry Streaming Services in SDN-Based Disaggregated Optical Networks”. en. In: *Journal of Lightwave Technology* 36.15 (Aug. 2018), pp. 3142–3149. ISSN: 0733-8724, 1558-2213. DOI: 10.1109/JLT.2018.2795345. URL: <https://ieeexplore.ieee.org/document/8263153/> (visited on 01/31/2022).
- [7] Weibo Liu et al. “A survey of deep neural network architectures and their applications”. en. In: *Neurocomputing* 234 (Apr. 2017), pp. 11–26. ISSN: 09252312. DOI: 10.1016/j.neucom.2016.12.038. URL: <https://linkinghub.elsevier.com/retrieve/pii/S0925231216315533> (visited on 12/01/2021).

-
- [8] Ignacio Martin et al. “Machine Learning-Based Routing and Wavelength Assignment in Software-Defined Optical Networks”. en. In: *IEEE Transactions on Network and Service Management* 16.3 (Sept. 2019), pp. 871–883. ISSN: 1932-4537, 2373-7379. DOI: 10.1109/TNSM.2019.2927867. URL: <https://ieeexplore.ieee.org/document/8758853/> (visited on 04/13/2021).
- [9] Matteo Salani, Cristina Rottondi, and Massimo Tornatore. “Routing and Spectrum Assignment Integrating Machine-Learning-Based QoT Estimation in Elastic Optical Networks”. en. In: *IEEE INFOCOM 2019 - IEEE Conference on Computer Communications*. Paris, France: IEEE, Apr. 2019, pp. 1738–1746. ISBN: 978-1-72810-515-4. DOI: 10.1109/INFOCOM.2019.8737413. URL: <https://ieeexplore.ieee.org/document/8737413/> (visited on 12/10/2019).
- [10] Yvan Pointurier. “Machine learning techniques for quality of transmission estimation in optical networks”. en. In: *Journal of Optical Communications and Networking* 13.4 (Apr. 2021), B60. ISSN: 1943-0620, 1943-0639. DOI: 10.1364/JOCN.417434. URL: <https://www.osapublishing.org/abstract.cfm?URI=jocn-13-4-B60> (visited on 03/25/2021).
- [11] Tarem Ahmed and Mark Coates. “Machine Learning Algorithms for Anomaly Detection in Optical Networks”. en. In: (), p. 1.
- [12] G. P Agrawal. *Fiber-optic Communication Systems (Wiley series in microwave and optical engineering)*. John Wiley & Sons Incorporated., 0. ISBN: 978-0-471-22114-2 978-0-471-17540-7.
- [13] Danshi Wang et al. “The Role of Digital Twin in Optical Communication: Fault Management, Hardware Configuration, and Transmission Simulation”. In: *IEEE Communications Magazine* 59.1 (2021), pp. 133–139. DOI: 10.1109/MCOM.001.2000727.
- [14] Reda Ayassi et al. “Survey on the Use of Machine Learning for Quality of Transmission Estimation in Optical Transport Networks”. In: *Journal of Lightwave Technology* (2022), pp. 1–13. ISSN: 0733-8724, 1558-2213. DOI: 10.1109/JLT.2022.3184178. URL: <https://ieeexplore.ieee.org/document/9799746/> (visited on 07/05/2022).
- [15] Jatin Babbar et al. “Machine learning models for alarm classification and failure localization in optical transport networks”. In: *J. Opt. Commun. Netw.* 14.8 (Aug. 2022), pp. 621–628. DOI: 10.1364/JOCN.457687. URL: <https://opg.optica.org/jocn/abstract.cfm?URI=jocn-14-8-621>.
- [16] Reda Ayassi et al. “Field Trial to Assess Bayesian Optimization for Improving QoT Estimation”. In: *49th European Conference on Optical Communications (ECOC 2023)*. 2023.
- [17] Reda Ayassi et al. “Bayesian optimization-based algorithm to improve the quality of transmission estimation”. en. In: (), p. 3.

-
- [18] R. Ayassi et al. “An Overview on Machine Learning-Based Solutions to Improve Lightpath QoT Estimation”. In: *2020 22nd International Conference on Transparent Optical Networks (ICTON)*. event-place: Bari, Italy. IEEE, July 2020, pp. 1–4. ISBN: 978-1-72818-423-4. DOI: 10.1109/ICTON51198.2020.9203755. URL: <https://ieeexplore.ieee.org/document/9203755/> (visited on 01/12/2021).
- [19] Clara Catanese et al. “A Fully Connected Neural Network to Mitigate 200G DP-16-QAM Transmission System Impairments”. en. In: *OSA Advanced Photonics Congress (AP) 2020 (IPR, NP, NOMA, Networks, PVLED, PSC, SPPCom, SOF)*. Washington, DC: OSA, 2020, SpTh3I.1. ISBN: 978-1-943580-79-8. DOI: 10.1364/SPPCOM.2020.SpTh3I.1. URL: <https://www.osapublishing.org/abstract.cfm?URI=SPPCom-2020-SpTh3I.1> (visited on 09/29/2021).
- [20] Clara Catanese et al. “A Fully Connected Neural Network Approach to Mitigate Fiber Nonlinear Effects in 200G DP-16-QAM Transmission System”. en. In: *2020 22nd International Conference on Transparent Optical Networks (ICTON)*. Bari, Italy: IEEE, July 2020, pp. 1–4. ISBN: 978-1-72818-423-4. DOI: 10.1109/ICTON51198.2020.9203197. URL: <https://ieeexplore.ieee.org/document/9203197/> (visited on 09/29/2021).
- [21] Clara Catanese et al. “A Survey of Neural Network Applications in Fiber Nonlinearity Mitigation”. en. In: *2019 21st International Conference on Transparent Optical Networks (ICTON)*. Angers, France: IEEE, July 2019, pp. 1–4. ISBN: 978-1-72812-779-8. DOI: 10.1109/ICTON.2019.8840355. URL: <https://ieeexplore.ieee.org/document/8840355/> (visited on 12/05/2019).
- [22] S. Donati and G. Giuliani. “Noise in an optical amplifier: formulation of a new semi-classical model”. en. In: *IEEE Journal of Quantum Electronics* 33.9 (Sept. 1997), pp. 1481–1488. ISSN: 00189197. DOI: 10.1109/3.622626. URL: <http://ieeexplore.ieee.org/document/622626/> (visited on 02/01/2022).
- [23] Daniel Garipey et al. “Novel OSNR Measurement Techniques Based on Optical Spectrum Analysis and Their Application to Coherent-Detection Systems”. en. In: *Journal of Lightwave Technology* 37.2 (Jan. 2019), pp. 562–570. ISSN: 0733-8724, 1558-2213. DOI: 10.1109/JLT.2018.2878744. URL: <https://ieeexplore.ieee.org/document/8515023/> (visited on 11/04/2021).
- [24] Yvan Pointurier. “Design of Low-Margin Optical Networks”. In: *J. Opt. Commun. Netw.* 9.1 (Jan. 2017), A9. ISSN: 1943-0620, 1943-0639. DOI: 10.1364/JOCN.9.0000A9. URL: <https://www.osapublishing.org/abstract.cfm?URI=jocn-9-1-A9> (visited on 04/08/2020).
- [25] Gert Grammel, Vittorio Curri, and Jean-Luc Auge. “Physical Simulation Environment of The Telecommunications Infrastructure Project (TIP)”. en. In: *Optical Fiber Communication Conference*. San Diego, California: OSA, 2018, p. M1D.3. ISBN: 978-1-943580-38-5. DOI: 10.1364/OFC.2018.M1D.3. URL: <https://www.osapublishing.org/abstract.cfm?URI=OFC-2018-M1D.3> (visited on 12/05/2019).

- [26] P. Poggiolini and Y. Jiang. “Recent Advances in the Modeling of the Impact of Non-linear Fiber Propagation Effects on Uncompensated Coherent Transmission Systems”. en. In: *Journal of Lightwave Technology* 35.3 (Feb. 2017), pp. 458–480. ISSN: 0733-8724, 1558-2213. DOI: 10.1109/JLT.2016.2613893. URL: <http://ieeexplore.ieee.org/document/7577767/> (visited on 11/18/2021).
- [27] Ya.I. Bogomolov and A.D. Yunakovsky. “Split-step Fourier method for nonlinear Schrodinger equation”. In: *DAYS on DIFFRACTION 2006*. St.Petersburg, Russia: IEEE, 2006, pp. 34–42. ISBN: 978-5-9651-0226-6. DOI: 10.1109/DD.2006.348170. URL: <http://ieeexplore.ieee.org/document/4154014/> (visited on 09/29/2021).
- [28] K.V. Peddanarappagari and M. Brandt-Pearce. “Volterra series transfer function of single-mode fibers”. In: *Journal of Lightwave Technology* 15.12 (Dec. 1997), pp. 2232–2241. ISSN: 07338724. DOI: 10.1109/50.643545. URL: <http://ieeexplore.ieee.org/document/643545/> (visited on 02/01/2022).
- [29] Marco Secondini, Enrico Forestieri, and Curtis R. Menyuk. “A Combined Regular-Logarithmic Perturbation Method for Signal-Noise Interaction in Amplified Optical Systems”. In: *Journal of Lightwave Technology* 27.16 (Aug. 2009), pp. 3358–3369. ISSN: 0733-8724. DOI: 10.1109/JLT.2009.2012873. URL: <http://ieeexplore.ieee.org/document/5170205/> (visited on 02/01/2022).
- [30] P. Poggiolini et al. “The GN-Model of Fiber Non-Linear Propagation and its Applications”. In: *J. Lightwave Technol.* 32.4 (Feb. 2014), pp. 694–721. ISSN: 0733-8724, 1558-2213. DOI: 10.1109/JLT.2013.2295208. URL: <http://ieeexplore.ieee.org/document/6685826/> (visited on 03/30/2020).
- [31] D. Marcuse, C.R. Menyuk, and P.K.A. Wai. “Application of the Manakov-PMD equation to studies of signal propagation in optical fibers with randomly varying birefringence”. In: *Journal of Lightwave Technology* 15.9 (Sept. 1997), pp. 1735–1746. ISSN: 07338724. DOI: 10.1109/50.622902. URL: <http://ieeexplore.ieee.org/document/622902/> (visited on 02/01/2022).
- [32] Andrea Carena et al. “EGN model of non-linear fiber propagation”. en. In: *Optics Express* 22.13 (June 2014), p. 16335. ISSN: 1094-4087. DOI: 10.1364/OE.22.016335. URL: <https://www.osapublishing.org/oe/abstract.cfm?uri=oe-22-13-16335> (visited on 11/18/2021).
- [33] Daniel Semrau, Robert I. Killey, and Polina Bayvel. “The Gaussian Noise Model in the Presence of Inter-Channel Stimulated Raman Scattering”. en. In: *Journal of Lightwave Technology* 36.14 (July 2018), pp. 3046–3055. ISSN: 0733-8724, 1558-2213. DOI: 10.1109/JLT.2018.2830973. URL: <https://ieeexplore.ieee.org/document/8351897/> (visited on 09/09/2021).
- [34] A. Nespola et al. “Extensive Fiber Comparison and GN-model Validation in Uncompensated Links using DAC-generated Nyquist-WDM PM-16QAM Channels”. en. In: *Optical Fiber Communication Conference/National Fiber Optic Engineers Conference 2013*. Anaheim, California: OSA, 2013, OTh3G.5. ISBN: 978-1-55752-962-6. DOI:

- 10.1364/OFC.2013.0Th3G.5. URL: <https://www.osapublishing.org/abstract.cfm?URI=OFC-2013-0Th3G.5> (visited on 11/18/2021).
- [35] Antonio Caballero et al. “Experimental demonstration of a cognitive quality of transmission estimator for optical communication systems”. In: (2012), p. 3.
- [36] Alan A. Diaz-Montiel et al. “Active Wavelength Load as a Feature for QoT Estimation Based on Support Vector Machine”. In: *ICC 2019 - 2019 IEEE International Conference on Communications (ICC)*. event-place: Shanghai, China. IEEE, May 2019, pp. 1–6. ISBN: 978-1-5386-8088-9. DOI: 10.1109/ICC.2019.8761369. URL: <https://ieeexplore.ieee.org/document/8761369/> (visited on 02/27/2020).
- [37] Tania Panayiotou et al. “Centralized and Distributed Machine Learning-Based QoT Estimation for Sliceable Optical Networks”. en. In: *arXiv:1908.08338 [cs, eess]* (Sept. 2019). arXiv: 1908.08338. URL: <http://arxiv.org/abs/1908.08338> (visited on 12/10/2019).
- [38] Stanisław Kozdrowski et al. “Machine Learning Algorithms for Prediction of the Quality of Transmission in Optical Networks”. en. In: *Entropy* 23.1 (Dec. 2020), p. 7. ISSN: 1099-4300. DOI: 10.3390/e23010007. URL: <https://www.mdpi.com/1099-4300/23/1/7> (visited on 04/19/2021).
- [39] Rui Manuel Morais and João Pedro. “Machine Learning Models for Estimating Quality of Transmission in DWDM Networks”. In: *J. Opt. Commun. Netw.* 10.10 (Oct. 2018), p. D84. ISSN: 1943-0620, 1943-0639. DOI: 10.1364/JOCN.10.000D84. URL: <https://www.osapublishing.org/abstract.cfm?URI=jocn-10-10-D84> (visited on 01/08/2020).
- [40] Cristina Rottondi et al. “Machine-Learning Method for Quality of Transmission Prediction of Unestablished Lightpaths”. In: *J. Opt. Commun. Netw.* 10.2 (Feb. 2018), A286. ISSN: 1943-0620, 1943-0639. DOI: 10.1364/JOCN.10.00A286. URL: <https://www.osapublishing.org/abstract.cfm?URI=jocn-10-2-A286> (visited on 02/12/2020).
- [41] Sandra Aladin et al. “Quality of Transmission Estimation and Short-Term Performance Forecast of Lightpaths”. en. In: *Journal of Lightwave Technology* 38.10 (May 2020), pp. 2807–2814. ISSN: 0733-8724, 1558-2213. DOI: 10.1109/JLT.2020.2975179. URL: <https://ieeexplore.ieee.org/document/9004503/> (visited on 04/12/2021).
- [42] Javier Mata et al. “Supervised Machine Learning Techniques for Quality of Transmission Assessment in Optical Networks”. In: *2018 20th International Conference on Transparent Optical Networks (ICTON)*. event-place: Bucharest. IEEE, July 2018, pp. 1–4. ISBN: 978-1-5386-6605-0. DOI: 10.1109/ICTON.2018.8473819. URL: <https://ieeexplore.ieee.org/document/8473819/> (visited on 12/10/2019).
- [43] Tania Panayiotou et al. “Machine Learning for QoT Estimation of Unseen Optical Network States”. In: *Optical Fiber Communication Conference (OFC) 2019*. event-place: San Diego, California. OSA, 2019, Tu2E.2. ISBN: 978-1-943580-53-8. DOI: 10.1364/OFC.2019.Tu2E.2. URL: <https://www.osapublishing.org/abstract.cfm?URI=OFC-2019-Tu2E.2> (visited on 02/10/2020).

- [44] Qirui Fan et al. “Experimental Comparisons between Machine Learning and Analytical Models for QoT Estimations in WDM Systems”. en. In: *Optical Fiber Communication Conference (OFC) 2020*. San Diego, California: OSA, 2020, M2J.2. ISBN: 978-1-943580-71-2. DOI: 10.1364/OFC.2020.M2J.2. URL: <https://www.osapublishing.org/abstract.cfm?URI=OFC-2020-M2J.2> (visited on 04/07/2021).
- [45] Zhengguang Gao et al. “ANN-Based Multi-Channel QoT-Prediction Over a 563.4-km Field-Trial Testbed”. en. In: *Journal of Lightwave Technology* 38.9 (May 2020), pp. 2646–2655. ISSN: 0733-8724, 1558-2213. DOI: 10.1109/JLT.2020.2971104. URL: <https://ieeexplore.ieee.org/document/8978723/> (visited on 04/13/2021).
- [46] Memedhe Ibrahimy et al. “Machine Learning Regression vs. Classification for QoT Estimation of Unestablished Lightpaths”. en. In: *OSA Advanced Photonics Congress (AP) 2020 (IPR, NP, NOMA, Networks, PVLED, PSC, SPPCom, SOF)*. Washington, DC: OSA, 2020, NeM3B.1. ISBN: 978-1-943580-79-8. DOI: 10.1364/NETWORKS.2020.NeM3B.1. URL: <https://www.osapublishing.org/abstract.cfm?URI=Networks-2020-NeM3B.1> (visited on 03/25/2021).
- [47] Ihtesham Khan, Muhammad Bilal, and Vittorio Curri. “Advanced Formulation of QoT-Estimation for Un-established Lightpaths Using Cross-train Machine Learning Methods”. en. In: *2020 22nd International Conference on Transparent Optical Networks (ICTON)*. Bari, Italy: IEEE, July 2020, pp. 1–4. ISBN: 978-1-72818-423-4. DOI: 10.1109/ICTON51198.2020.9203334. URL: <https://ieeexplore.ieee.org/document/9203334/> (visited on 09/08/2021).
- [48] Ihtesham Khan et al. “QoT Estimation for Light-path Provisioning in Un-Seen Optical Networks using Machine Learning”. en. In: *2020 22nd International Conference on Transparent Optical Networks (ICTON)*. Bari, Italy: IEEE, July 2020, pp. 1–4. ISBN: 978-1-72818-423-4. DOI: 10.1109/ICTON51198.2020.9203364. URL: <https://ieeexplore.ieee.org/document/9203364/> (visited on 04/13/2021).
- [49] F. Meng et al. “Field Trial of Gaussian Process Learning of Function-Agnostic Channel Performance Under Uncertainty”. In: *Optical Fiber Communication Conference*. event-place: San Diego, California. OSA, 2018, W4F.5. ISBN: 978-1-943580-38-5. DOI: 10.1364/OFC.2018.W4F.5. URL: <https://www.osapublishing.org/abstract.cfm?URI=OFC-2018-W4F.5> (visited on 02/04/2020).
- [50] Shuangyi Yan et al. “Field trial of Machine-Learning-assisted and SDN-based Optical Network Planning with Network-Scale Monitoring Database”. In: *2017 European Conference on Optical Communication (ECOC)*. Gothenburg: IEEE, Sept. 2017, pp. 1–3. ISBN: 978-1-5386-5624-2. DOI: 10.1109/ECOC.2017.8346091. URL: <https://ieeexplore.ieee.org/document/8346091/> (visited on 04/07/2021).
- [51] Jasper Muller et al. “Estimating Quality of Transmission in a Live Production Network using Machine Learning”. en. In: (2021), p. 3.

-
- [52] R. Proietti et al. “Experimental Demonstration of Cognitive Provisioning and Alien Wavelength Monitoring in Multi-domain EON”. In: *Optical Fiber Communication Conference*. event-place: San Diego, California. OSA, 2018, W4F.7. ISBN: 978-1-943580-38-5. DOI: 10.1364/OFC.2018.W4F.7. URL: <https://www.osapublishing.org/abstract.cfm?URI=OFC-2018-W4F.7> (visited on 02/17/2020).
- [53] Payman Samadi et al. “Quality of Transmission Prediction with Machine Learning for Dynamic Operation of Optical WDM Networks”. en. In: *2017 European Conference on Optical Communication (ECOC)*. Gothenburg: IEEE, Sept. 2017, pp. 1–3. ISBN: 978-1-5386-5624-2. DOI: 10.1109/ECOC.2017.8346216. URL: <https://ieeexplore.ieee.org/document/8346216/> (visited on 03/25/2021).
- [54] Fehmida Usmani et al. “Convolutional neural network for quality of transmission prediction of unestablished lightpaths”. en. In: *Microwave and Optical Technology Letters* 63.10 (Oct. 2021), pp. 2461–2469. ISSN: 0895-2477, 1098-2760. DOI: 10.1002/mop.32996. URL: <https://onlinelibrary.wiley.com/doi/10.1002/mop.32996> (visited on 09/08/2021).
- [55] Jesper Wass et al. “Gaussian Process Regression for WDM System Performance Prediction”. en. In: *Optical Fiber Communication Conference*. Los Angeles, California: OSA, 2017, Tu3D.7. ISBN: 978-1-943580-23-1. DOI: 10.1364/OFC.2017.Tu3D.7. URL: <https://www.osapublishing.org/abstract.cfm?URI=OFC-2017-Tu3D.7> (visited on 04/13/2021).
- [56] Fehmida Usmani et al. “Evaluating Cross- feature Trained Machine Learning Models for Estimating QoT of Unestablished Lightpaths”. en. In: *2021 International Conference on Electrical, Communication, and Computer Engineering (ICECCE)*. Kuala Lumpur, Malaysia: IEEE, June 2021, pp. 1–6. ISBN: 978-1-66543-897-1. DOI: 10.1109/ICECCE52056.2021.9514154. URL: <https://ieeexplore.ieee.org/document/9514154/> (visited on 09/09/2021).
- [57] Emmanuel Seve, Jelena Pesic, and Yvan Pointurier. “Associating machine-learning and analytical models for quality of transmission estimation: combining the best of both worlds”. en. In: *Journal of Optical Communications and Networking* 13.6 (June 2021), p. C21. ISSN: 1943-0620, 1943-0639. DOI: 10.1364/JOCN.411979. URL: <https://www.osapublishing.org/abstract.cfm?URI=jocn-13-6-C21> (visited on 04/13/2021).
- [58] Nathalie Morette. “Leveraging ML-based QoT Tool Parameter Feeding for Accurate WDM Network Performance Prediction”. en. In: (2021), p. 3.
- [59] Ippokratis Sartzetakis, Konstantinos (Kostas) Christodouloupoulos, and Emmanouel (Manos) Varvarigos. “Accurate Quality of Transmission Estimation With Machine Learning”. en. In: *Journal of Optical Communications and Networking* 11.3 (Mar. 2019), p. 140. ISSN: 1943-0620, 1943-0639. DOI: 10.1364/JOCN.11.000140. URL: <https://www.osapublishing.org/abstract.cfm?URI=jocn-11-3-140> (visited on 12/10/2019).

-
- [60] E. Seve et al. “Learning Process for Reducing Uncertainties on Network Parameters and Design Margins”. In: *J. Opt. Commun. Netw.* 10.2 (Feb. 2018), A298. ISSN: 1943-0620, 1943-0639. DOI: 10.1364/JOCN.10.00A298. URL: <https://www.osapublishing.org/abstract.cfm?URI=jocn-10-2-A298> (visited on 12/10/2019).
- [61] Martin Bouda, Oda Shoichiro, and Vasilieva Olga. “Accurate prediction of quality of transmission with dynamically configurable optical impairment model”. In: Mar. 2017.
- [62] Fanchao Meng et al. “Robust Self-Learning Physical Layer Abstraction Utilizing Optical Performance Monitoring and Markov Chain Monte Carlo”. In: *2017 European Conference on Optical Communication (ECOC)*. event-place: Gothenburg. IEEE, Sept. 2017, pp. 1–3. ISBN: 978-1-5386-5624-2. DOI: 10.1109/ECOC.2017.8346217. URL: <https://ieeexplore.ieee.org/document/8346217/> (visited on 03/12/2020).
- [63] Andrea D’Amico et al. “Using Machine-Learning in an Open Optical Line Systems Controller”. In: (2019), p. 9.
- [64] Qunbi Zhuge and Weisheng Hu. “Application of Machine Learning in Elastic Optical Networks”. In: *2018 European Conference on Optical Communication (ECOC)*. event-place: Rome. IEEE, Sept. 2018, pp. 1–3. ISBN: 978-1-5386-4862-9. DOI: 10.1109/ECOC.2018.8535565. URL: <https://ieeexplore.ieee.org/document/8535565/> (visited on 02/04/2020).
- [65] Rui Manuel Morais, Bruno Pereira, and João Pedro. “Fast and High-Precision Optical Performance Evaluation for Cognitive Optical Networks”. en. In: *Optical Fiber Communication Conference (OFC) 2020*. San Diego, California: OSA, 2020, Th3D.3. ISBN: 978-1-943580-71-2. DOI: 10.1364/OFC.2020.Th3D.3. URL: <https://www.osapublishing.org/abstract.cfm?URI=OFC-2020-Th3D.3> (visited on 04/13/2021).
- [66] Ankush Mahajan et al. “Modeling Filtering Penalties in ROADMs-based Networks with Machine Learning for QoT Estimation”. In: *Optical Fiber Communication Conference (OFC) 2020*. event-place: San Diego, California. OSA, 2020, Th3D.4. ISBN: 978-1-943580-71-2. DOI: 10.1364/OFC.2020.Th3D.4. URL: <https://www.osapublishing.org/abstract.cfm?URI=OFC-2020-Th3D.4> (visited on 03/21/2020).
- [67] D Uzunidis et al. “Enhancing Closed-Form Based Physical Layer Performance Estimations in EONs Via Machine Learning Techniques”. en. In: (), p. 4.
- [68] Alessio Ferrari et al. “Experimental Validation of an Open Source Quality of Transmission Estimator for Open Optical Networks”. In: *Optical Fiber Communication Conference (OFC) 2020*. event-place: San Diego, California. OSA, 2020, W3C.2. ISBN: 978-1-943580-71-2. DOI: 10.1364/OFC.2020.W3C.2. URL: <https://www.osapublishing.org/abstract.cfm?URI=OFC-2020-W3C.2> (visited on 03/20/2020).
- [69] Shuang Yao et al. “Data Efficient Estimation for Quality of Transmission Through Active Learning in Fiber-Wireless Integrated Network”. en. In: *Journal of Lightwave Technology* 39.18 (Sept. 2021), pp. 5691–5698. ISSN: 0733-8724, 1558-2213. DOI: 10.1109/JLT.2021.3091377. URL: <https://ieeexplore.ieee.org/document/9462465/> (visited on 09/30/2021).

-
- [70] Che-Yu Liu et al. “Performance studies of evolutionary transfer learning for end-to-end QoT estimation in multi-domain optical networks [Invited]”. en. In: *Journal of Optical Communications and Networking* 13.4 (Apr. 2021), B1. ISSN: 1943-0620, 1943-0639. DOI: 10.1364/JOCN.409817. URL: <https://www.osapublishing.org/abstract.cfm?URI=jocn-13-4-B1> (visited on 04/13/2021).
- [71] Ihtesham Khan et al. “Lightpath QoT computation in optical networks assisted by transfer learning”. en. In: *Journal of Optical Communications and Networking* 13.4 (Apr. 2021), B72. ISSN: 1943-0620, 1943-0639. DOI: 10.1364/JOCN.409538. URL: <https://www.osapublishing.org/abstract.cfm?URI=jocn-13-4-B72> (visited on 04/13/2021).
- [72] Dario Azzimonti et al. “Comparison of domain adaptation and active learning techniques for quality of transmission estimation with small-sized training datasets [Invited]”. en. In: *Journal of Optical Communications and Networking* 13.1 (Jan. 2021), A56. ISSN: 1943-0620, 1943-0639. DOI: 10.1364/JOCN.401918. URL: <https://www.osapublishing.org/abstract.cfm?URI=jocn-13-1-A56> (visited on 04/13/2021).
- [73] Jelena Pesic et al. “Transfer Learning from Unbiased Training Data Sets for QoT Estimation in WDM Networks”. en. In: *2020 European Conference on Optical Communications (ECOC)*. Brussels, Belgium: IEEE, Dec. 2020, pp. 1–4. ISBN: 978-1-72817-361-0. DOI: 10.1109/ECOC48923.2020.9333399. URL: <https://ieeexplore.ieee.org/document/9333399/> (visited on 04/13/2021).
- [74] Jiakai Yu et al. “Model transfer of QoT prediction in optical networks based on artificial neural networks”. In: *J. Opt. Commun. Netw.* 11.10 (Oct. 2019), p. C48. ISSN: 1943-0620, 1943-0639. DOI: 10.1364/JOCN.11.000C48. URL: <https://www.osapublishing.org/abstract.cfm?URI=jocn-11-10-C48> (visited on 01/08/2020).
- [75] Cristina Rottondi et al. “On the benefits of domain adaptation techniques for quality of transmission estimation in optical networks”. en. In: *Journal of Optical Communications and Networking* 13.1 (Jan. 2021), A34. ISSN: 1943-0620, 1943-0639. DOI: 10.1364/JOCN.401915. URL: <https://www.osapublishing.org/abstract.cfm?URI=jocn-13-1-A34> (visited on 04/13/2021).
- [76] Le Xia et al. “Transfer learning assisted deep neural network for OSNR estimation”. en. In: *Optics Express* 27.14 (July 2019), p. 19398. ISSN: 1094-4087. DOI: 10.1364/OE.27.019398. URL: <https://www.osapublishing.org/abstract.cfm?URI=oe-27-14-19398> (visited on 04/13/2021).
- [77] Matteo Lonardi et al. “The Perks of Using Machine Learning for QoT Estimation with Uncertain Network Parameters”. en. In: *OSA Advanced Photonics Congress (AP) 2020 (IPR, NP, NOMA, Networks, PVLED, PSC, SPPCom, SOF)*. Washington, DC: OSA, 2020, NeM3B.2. ISBN: 978-1-943580-79-8. DOI: 10.1364/NETWORKS.2020.NeM3B.2. URL: <https://www.osapublishing.org/abstract.cfm?URI=Networks-2020-NeM3B.2> (visited on 03/25/2021).

- [78] Alessio Ferrari et al. “Assessment on the in-field lightpath QoT computation including connector loss uncertainties”. In: *J. Opt. Commun. Netw.* 13.2 (Feb. 2021), A156. ISSN: 1943-0620, 1943-0639. DOI: 10.1364/JOCN.402969. URL: <https://www.osapublishing.org/abstract.cfm?URI=jocn-13-2-A156> (visited on 01/01/2021).
- [79] K. Christodoulopoulos et al. “Toward efficient, reliable, and autonomous optical networks: the ORCHESTRA solution [Invited]”. In: *J. Opt. Commun. Netw.* 11.9 (Sept. 2019), p. C10. ISSN: 1943-0620, 1943-0639. DOI: 10.1364/JOCN.11.000C10. URL: <https://www.osapublishing.org/abstract.cfm?URI=jocn-11-9-C10> (visited on 02/04/2020).
- [80] Alessio Ferrari et al. “GNPy: an open source application for physical layer aware open optical networks”. en. In: *Journal of Optical Communications and Networking* 12.6 (June 2020), p. C31. ISSN: 1943-0620, 1943-0639. DOI: 10.1364/JOCN.382906. URL: <https://www.osapublishing.org/abstract.cfm?URI=jocn-12-6-C31> (visited on 01/25/2021).
- [81] Jianing Lu et al. “Automated training dataset collection system design for machine learning application in optical networks: an example of quality of transmission estimation”. en. In: *Journal of Optical Communications and Networking* 13.11 (Nov. 2021), p. 289. ISSN: 1943-0620, 1943-0639. DOI: 10.1364/JOCN.431780. URL: <https://www.osapublishing.org/abstract.cfm?URI=jocn-13-11-289> (visited on 12/16/2021).
- [82] Alessio Ferrari et al. “GNPy: an open source application for physical layer aware open optical networks”. In: *J. Opt. Commun. Netw.* 12.6 (June 2020), p. C31. ISSN: 1943-0620, 1943-0639. DOI: 10.1364/JOCN.382906. URL: <https://www.osapublishing.org/abstract.cfm?URI=jocn-12-6-C31> (visited on 01/25/2021).
- [83] Diederik P. Kingma and Jimmy Ba. *Adam: A Method for Stochastic Optimization*. 2017. arXiv: 1412.6980 [cs.LG].
- [84] Jong-Wan Kim and Chang-Hee Lee. “Modulation format identification of square and non-square M-QAM signals based on amplitude variance and OSNR”. In: *Optics Communications* 474 (2020), p. 126084. ISSN: 0030-4018. DOI: <https://doi.org/10.1016/j.optcom.2020.126084>. URL: <https://www.sciencedirect.com/science/article/pii/S0030401820305010>.
- [85] Zhao Zhao, Aiying Yang, and Peng Guo. “A Modulation Format Identification Method Based on Information Entropy Analysis of Received Optical Communication Signal”. In: *IEEE Access* 7 (2019), pp. 41492–41497. DOI: 10.1109/ACCESS.2019.2907521.
- [86] Emmanuel Seve et al. “Automated Fiber Type Identification in SDN-Enabled Optical Networks”. In: *Journal of Lightwave Technology* 37.7 (2019), pp. 1724–1731. DOI: 10.1109/JLT.2019.2896041.
- [87] Bobak Shahriari et al. “Taking the human out of the loop: A review of Bayesian optimization”. In: *Proceedings of the IEEE* 104.1 (2015), pp. 148–175.

-
- [88] Oriol Vinyals et al. “Matching Networks for One Shot Learning”. In: *Advances in Neural Information Processing Systems*. Ed. by D. Lee et al. Vol. 29. Curran Associates, Inc., 2016. URL: https://proceedings.neurips.cc/paper_files/paper/2016/file/90e1357833654983612fb05e3ec9148c-Paper.pdf.
- [89] Volodymyr Mnih et al. “Human-level control through deep reinforcement learning”. en. In: *Nature* 518.7540 (Feb. 2015), pp. 529–533. ISSN: 0028-0836, 1476-4687. DOI: 10.1038/nature14236. URL: <http://www.nature.com/articles/nature14236> (visited on 09/29/2021).
- [90] Jianqing Fan et al. “A Theoretical Analysis of Deep Q-Learning”. In: *Proceedings of the 2nd Conference on Learning for Dynamics and Control*. Ed. by Alexandre M. Bayen et al. Vol. 120. Proceedings of Machine Learning Research. PMLR, Oct. 2020, pp. 486–489. URL: <https://proceedings.mlr.press/v120/yang20a.html>.
- [91] Seiji Okamoto et al. “A Study on the Effect of Ultra-Wide Band WDM on Optical Transmission Systems”. In: *J. Lightwave Technol.* 38.5 (Mar. 2020), pp. 1061–1070. URL: <https://opg.optica.org/jlt/abstract.cfm?URI=jlt-38-5-1061>.
- [92] M.N. Islam. “Raman amplifiers for telecommunications”. In: *IEEE Journal of Selected Topics in Quantum Electronics* 8.3 (2002), pp. 548–559. DOI: 10.1109/JSTQE.2002.1016358.

List of figures

2.1	General process to train a ML QoT estimation model	31
2.2	Learning process to reduce uncertainty on parameters	35
2.3	Learning process to adapt a ML model from one domain to another using Transfer Learning	39
2.4	Four levels of data features for ML-based solutions	43
3.1	QoT computation and improvement framework	48
3.2	Structure of the Neural Network Based QoT Model	52
3.3	Network Topology Used to Generate the Training Dataset for the NN Model	53
3.4	Evolution of the training loss during model training	54
3.5	Network Topology Used to Calculate Parameter Impact	55
3.6	Maximum SNR error generated by gradually changing the NF of each amplifier in the 4 OMS FLAT_NET topology	56
3.7	Mean SNR error generated by gradually changing the NF of each amplifier in the 4 OMS FLAT_NET topology	57
3.8	Mean SNR error generated by gradually changing the CONN_OUT loss in each span in the 4 OMS FLAT_NET topology in power mode	58
3.9	Mean SNR error generated by gradually changing the CONN_OUT loss in each span in the 4 OMS FLAT_NET topology in gain mode	59
3.10	SNR spectrum generated by changing the CONN_IN loss in each span in the 1 OMS FLAT_NET topology	59
4.1	The FLAT_NET simulation topology	65
4.2	Bayesian optimization algorithm process. Figure taken from [87]	68
4.3	LP1 - Simulation 1 Results: SNR spectrum in each network state during the learning process	71
4.4	LP1 - Simulation 1 Results: Error between the target and best values for each uncertain parameter in the learning process	72
4.5	LP1 - simulation 2 results: SNR spectrum in each network state during the learning process	73
4.6	LP1 - simulation 2 results: Error between the target and best values for each uncertain parameter in the learning process	74
4.7	LP1 - simulation 3 results: SNR spectrum for each network state during the learning process	75
4.8	Gradient descent algorithm process	77
4.9	LP2 - simulation 1 results: comparison between the SNR estimation of GNP _y and that of NN	78
4.10	LP2 - simulation 2 results: SNR spectrum for each network state during the learning process	80
4.11	LP2 - simulation 2 results: Error between the target and best values for each uncertain parameter in the learning process	81
4.12	LP2 - simulation 3 results: SNR spectrum comparison between GNP _y and NN model before and after the optimization	82
4.13	LP2 - Simulation 3 results: SNR spectrum for each network state during the power shift optimization	83
4.14	LP3 simulation 1: Dataset composition used to retrain the NN model	86
4.15	LP3 simulation results 1: training and validation results with TRAIN_PRIO method	88
4.16	LP3 simulation results 1: training and validation results with VAL_PRIO method	88
4.17	LP3 simulation results 2: Dataset composition used to retrain the NN model	89
4.18	LP3 simulation results 2: training and validation results with VAL_PRIO method	90
4.19	Structure of the reinforcement learning model using in LP4	93

4.20 LP4 Simulation 1 results: Evolution of the EV_ACC and QoT_ERR at different milestones in the training phase	96
4.21 LP4 Simulation 1 results: Validation results	97
5.1 The sub-network transmission lines used for the optimization process	104
5.2 Progressive deployment scenario - Mean SNR error progression for Line 1.	106
5.3 Progressive deployment scenario - SNR error per service for Line 1.	106
5.4 Progressive deployment scenario - Mean SNR error progression for Line 2.	107
5.5 Progressive deployment scenario - SNR error per service for Line 2.	107
6.1 Nonlinear effects in optical transmission	134

List of tables

2.1	Comparison of papers proposing a ML model to check lightpath feasibility	33
2.2	Comparison of papers proposing a ML model to estimate a lightpath's QoT metric	34
2.3	Comparison of papers attempting to enhance analytical models through machine learning	36
2.4	Comparison of papers featuring machine learning techniques for dataset management	38
2.5	Four levels of data features for ML-based solutions	44
3.1	List of Parameters Used to Generate the Training and Validation Datasets	53
3.2	Summary of the parameters constituting the GNPY based network state and their level of certainty . .	62
5.1	List of features extracted from the NMS	102
5.2	Network state from extracted parameters	103
5.3	List of monitored services in the two transmission lines	105

Appendix

Optical Impairments

An optical network is composed of several nodes (optical equipment that are able to emit/receive and/or switch optical signal) inter-connected by fibers. Once emitted by a source node, data flows travel through the optical network without undergoing Optical-to-Electrical-to-Optical conversion at intermediate nodes, until arriving to the destination. The path followed by the data flow is called “lightpath”. The lightpath is typically transported by one optical wavelength. An optical fiber can multiplex more than 80 wavelengths. As optical signals traverse fiber links and nodes, and propagate through several optical components towards their destination, they suffer from a number of physical impairments which degrade the signal quality. These impairments affect each optical channel individually, and they also cause disturbance and interference between co-propagating channels. In this section, we make the inventory of the transmission impairments and the metrics to assess the quality of the optical signal.

Linear Effects

Linear effects mainly include fiber attenuation and dispersion effects.

Fiber Attenuation

Fiber attenuation measures the amount of light lost between input and output. The fiber loss along a span is given by equation 6.1.

$$SL = P_{in} - P_{out} \quad (6.1)$$

Where P_{in} and P_{out} present respectively the input and the output power in dBm. The attenuation coefficient (also called lineic loss) α is a characteristic of the fiber that shows optical loss (in dB) per kilometer. The relationship between the α and SL are given by equation 6.2, where z is the length of the span in km.

$$SL = z \cdot \alpha \quad (6.2)$$

The fiber loss is compensated using amplifiers that are located in strategic points (i.e., end of spans) along a long transmission link (e.g., several hundreds of kilometers) to restore the signal. The amplification process is done optically without converting the signal to the electric domain. The amplifier is basically characterized by a gain G . In the case of an amplifier i located in the end of the span $i - 1$ characterized by a span loss equal to SL_{i-1} , the gain of the amplifier G_i is given by equation 6.3.

$$G_i = SL_{i-1} + P_i - P_{i-1} \quad (6.3)$$

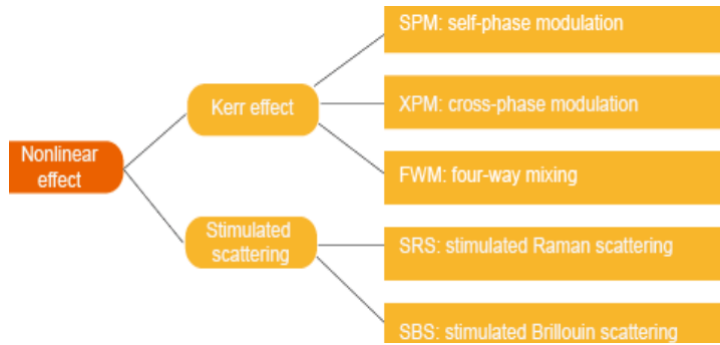


Figure 6.1: Nonlinear effects in optical transmission

Where P_i and P_{i-1} are the output power of the amplifier i and $i-1$. The amplification process causes additional noise to the signal called Amplified Spontaneous Emission (ASE) noise. The ASE noise degrades the transmission quality (see 6.2).

Dispersion Effects

Dispersion effects are caused by a difference in propagation speed between different light components. The most important kinds of dispersion are:

- the chromatic dispersion (CD): wavelengths of the same fiber propagate with different phase velocity,
- the polarization mode dispersion (PMD): the two different polarization of light travel in different speeds.

Dispersion is either compensated continuously throughout the line, or at reception by the DSP (Digital Signal Processor) module.

Nonlinear Effects

Nonlinear (NL) effects are by nature much harder to predict than linear ones. Transmitting at high rates and power levels causes multiple interference effects that stem from the variation of the fiber's refractive index and the interaction between optical channels. This causes multiple physical effects to manifest on the transmission line. Figure 6.1 gives a brief inventory of optical nonlinear effects. Optical transmission is primarily affected by two types of NLI impairments: i) the Raman scattering and ii) the Kerr effect. In [20], the mitigation of nonlinear effects has been studied in conjecture with AI techniques.

Raman Scattering

Raman scattering involves both, Spontaneous Raman Scattering and Simulated Raman Scattering. It occurs when several wavelengths are propagated through one fiber. The higher wavelength (the lower frequencies) can "absorb" energy from the lower wavelength (the higher frequencies). Thus, the higher wavelengths are amplified. Microscopically speaking, when a photon at energy level E_1 collides with an atom of the fiber, the atom absorbs the photon, becomes excited and emits another photon with a lower energy E_2 . This phenomenon causes a shift of the energy toward lower frequencies according to the formula in 6.4.

$$E_1 = h \cdot f_1 > E_2 = h \cdot f_2 \Rightarrow f_1 > f_2 \Rightarrow \lambda_1 < \lambda_2 \quad (6.4)$$

where E is the energy of the photon, h is the Planck constant and f is the frequency of the photon. Raman scattering represented the fundamental principle of Raman amplifiers (Raman pump).

Kerr Effects

The Kerr effect is a variation of the refractive index of the fiber due to the high power of the propagated signal. The Kerr-nonlinearity manifests itself in three different effects : SPM, XPM and FWM.

- Self-Phase Modulation (SPM):

SPM arises due to intensity dependence of refractive index. SPM means that the light wave encounters a nonlinear phase delay which results from its own intensity. The refractive index is intensity dependent, itself time dependent. Thus, the refractive index is time dependent. This temporally varying index change results in a temporally varying phase change. Let's consider refractive index n split between a linear portion n_l and a non-linear portion n_{nl} which is intensity I dependent as illustrate in the equation 6.5:

$$n(I) = n_l + n_{nl} \cdot I \quad (6.5)$$

The instantaneous phase Φ is time dependent and given by the equation 6.6.

$$\Phi(t, L_{eff}) = \omega_0 \cdot t - \frac{2\Pi}{\lambda} \cdot n(I)L_{eff} = \omega_0 \cdot t - \frac{2\Pi}{\lambda} \cdot (n_l + n_{nl}I)L_{eff} \quad (6.6)$$

where, ω_0 is the carrier frequency, L_{eff} is the effective length, n is the refractive index, I is intensity and n_{nl} is the non-linear refractive index.

If I is time dependent, the phase Φ will also be time dependent. In high transmitted power, the shift in intensity causes a shift in the phase and then in the instantaneous frequency as shown in equation 6.7 :

$$\omega(t) = \frac{\partial\Phi(t, L_{eff})}{\partial t} = \omega_0 - \frac{2\Pi}{\lambda} \cdot L_{eff} \cdot n_{nl} \cdot \frac{dI}{dt} \quad (6.7)$$

- Cross-Phase Modulation (XPM):

CPM is a non-linear impairment caused by the interaction of two (or more) laser pulses. It is always accompanied by SPM. When two or more optical pulses propagate simultaneously, the nonlinear refractive index depends not only on the intensity of one beam but also on the intensity of the other copropagating beams.

- Four-Wave Mixing (FWM):

The FWM is an analogous to intermodulation distortion in electrical systems. The effects of FWM are increased when channel spacing decreases. In facts, 2 or 3 wavelengths interact to generate a fourth wavelength. That is why it is called Four-Wave Mixing. For three pulses with carrier pulsations ω_1 , ω_2 and ω_3 copropagate inside the fiber simultaneously, a fourth will be generated at the pulsation ω_4 , which is related to other frequencies by a relation, $\omega_4 = \omega_1 \pm \omega_2 \pm \omega_3$.

GNLI Model

The Gaussian Noise (GN) model [30] is a perturbative model praised for being relatively simple while still providing adequate results. It offers an approximate analytical solution to the Schrodinger equation. The GN model has found great success in recent optical systems, due to the advent of Coherent-Detection and the use of uncompensated transmission techniques, that make the model's assumptions more plausible.

The GN model is based on three assumptions [30]:

- The nonlinear effects are small enough to be modeled as perturbations,
- The transmitted signal behaves as stationary Gaussian noise,
- The nonlinear disturbance is manifested as Additive Gaussian white noise.

This leads to multiple conclusions, namely the GN Reference Formula (GNRF) [30] in its multiple forms, that models the power spectral density of the nonlinear effects.

Optical Metrics

Amplified Spontaneous Emission (P_{ASE})

It represents the power level of the noise emitted by amplifiers. It depends on the noise figure and gain levels of the amplifier. It can be calculated at the output of each amplifier. Equations 6.8 and 6.9 calculate the linear and decibel P_{ASE} , respectively.

$$P_{ASE} = NF_{eq} \cdot h\nu\Delta\nu \cdot G \quad (6.8)$$

$$P_{ASE_{dbm}} = NF_{eq_{dbm}} + G_{dbm} - 58 \quad (6.9)$$

Where NF_{eq} is the amplifier noise factor and G is the amplifier gain. The ASE noise is accumulated from multiple amplifiers throughout the line.

Optical Signal to Noise Ratio (OSNR)

It represents the ratio between transmitted signal and noise generated by linear effects. Since most of the linear noise is generated by amplifier noise, OSNR is given by equation 6.10.

$$OSNR = \frac{P_{ch}}{P_{ASE}} \quad (6.10)$$

P_{ch} is the average power per channel, and P_{ASE} is the power of the Amplified Spontaneous Emission (ASE) noise. OSNR can be measured at the receiver side using a spectrometer. It is also easy to calculate analytically in the line when the amplifier noise factor is known.

Signal to Noise Ratio (SNR)

OSNR might be an indicator of optical quality, but it does not take into account the nonlinear effects. Instead, the SNR, which is the signal to noise ratio, takes into accounts all the transmission impairments.

$$SNR^{-1} = \frac{\sigma_{ASE}^2}{p} + \sigma_{TRx}^2 + \eta_{NL} \cdot p^2 \quad (6.11)$$

Where σ_{ASE}^2 , σ_{TRx}^2 and $\eta_{NL} \cdot p^2$ model the ASE, transceiver, and nonlinear noise, respectively. SNR cannot be effectively measured on a spectrometer since the nonlinear (NL) noise is spectrally superimposed to the signal spectrum. SNR can however be estimated either using a related measure eSNR [3], or by using a back to back technique to fit $SNR = f(BER)$ (explained further).

Taking into account the Gaussian additive nature of ASE and NLI noise, and considering transponder noise negligible, the SNR can be expressed by equation 6.12.

$$SNR = \frac{P_{ch}}{P_{ASE} + P_{NLI}} \quad (6.12)$$

P_{NLI} represents the power of nonlinear effects. It can be expressed in function of channel power by the equation 6.13.

$$P_{NLI} = A_{nl} \cdot P_{ch}^3 \quad (6.13)$$

A_{nl} is a constant (in regards to power) that characterizes the non-linear nature of the transmission and depends on multiple factors. This value can be computed using GNRF, or it can be calculated given real measurements of SNR and OSNR through equation 6.14.

$$A_{nl} = \left(\frac{1}{SNR} - \frac{1}{OSNR} \right) \cdot \frac{1}{P_{ch}^3} \quad (6.14)$$

Bit Error Rate (BER)

It represents the number of erroneous bits per second. It is an effective measure of the quality of transmission. The transponder can correct all errors up until a BER threshold, corresponding to the FEC (Forward Error Correction) limit. BER can be measured at receiver before (Pre-FEC BER) or after (Post-FEC BER) performing the FEC.

Q-Factor

It represents a measure of the signal quality, and is directly linked to BER using the equation 6.15.

$$BER = \frac{1}{2} \operatorname{erfc}\left(\frac{Q}{\sqrt{2}}\right) \simeq \frac{1}{\sqrt{2\pi}Q} \exp\left(-\frac{Q^2}{2}\right) \quad (6.15)$$

Q factor can be measured by the transceiver using eye diagrams. However, it is usually calculated from BER measurements. Using a back to back technique, we can find the relationship between SNR and Q factor.

Titre : Techniques Intelligence Artificielle pour l'Amélioration de l'Estimation de la Qualité de Transmission dans les Réseaux Optiques

Mots clés : Qualité de Transmission, Machine Learning, Intelligence Artificielle, Optimisation des Paramètres, Réseaux Optiques.

Résumé : L'estimation de la qualité de transmission (QoT) des chemins optiques est cruciale dans la conception du réseau et le provisionnement des services. Des études récentes se sont tournées vers les techniques de l'intelligence artificielle (IA) pour améliorer la précision de l'estimation de la QoT, en utilisant les données générées par le réseau optique. Nous distinguons quatre catégories de solutions. La première catégorie consiste à construire un modèle d'IA pour vérifier la faisabilité d'un chemin optique. La deuxième catégorie vise à proposer des modèles basés sur l'IA pour remplacer les modèles analytiques. La troisième catégorie utilise l'IA pour améliorer les performances des modèles d'estimation de la QoT en réduisant l'incertitude sur les paramètres d'entrée. La dernière catégorie consiste à améliorer les performances et la capacité de généralisation des solutions à base d'IA en améliorant les échantillons des jeux de données dans la phase d'apprentissage grâce à des techniques d'apprentissage par transfert. Les modèles d'estimation de la QoT peuvent constituer un module dans le Digital Twin du réseau optique, visant à simuler l'impact d'une nouvelle configuration sur la performance du réseau avant la phase de déploiement. Cependant, ces modèles requièrent une connaissance parfaite de l'état du réseau, représenté à partir d'un ensemble de paramètres optiques ayant des valeurs qui peuvent être certaines ou incertaines. Les mesures de performance collectées par le contrôleur peuvent représenter un feedback sur la précision de l'estimation de la QoT, ce qui peut déclencher des algorithmes à base de machine learning pour raffiner les valeurs des paramètres incertains. Dans cette thèse, nous étudions le problème d'incertitude des paramètres, et nous proposons trois approches pour améliorer la QoT dans ce cas. Nous proposons pour chaque approche un certain nombre de processus d'apprentissage

et nous testons leurs performances avec des données de simulation et des données collectées à partir du réseau opérationnel.

La première approche se base sur l'optimisation des paramètres du réseau en utilisant l'erreur dans l'estimation de la QoT comme fonction d'objectif. Nous implémentons cette approche avec deux processus d'apprentissage, le premier basé sur un modèle analytique (GNPy) et le deuxième sur un modèle à base de machine learning (réseau de neurones). Cette approche arrive à minimiser l'erreur d'estimation jusqu'à 0~dB pour des configurations de réseau où le modèle a été déjà entraîné, et atteint une erreur d'estimation de 0.3~dB sur des nouvelles configurations.

Dans la deuxième approche, nous ré-entraînons un modèle basé sur un réseau de neurones pour l'adapter à l'incertitude des paramètres en utilisant l'apprentissage par transfert. Nous montrons que le modèle peut apprendre un nouveau comportement sans optimiser les paramètres incertains. Nous arrivons à une erreur de validation de 0.5~dB avec seulement dix nouveaux échantillons.

La dernière approche consiste à détecter les changements de paramètres en réponse à un événement de panne en utilisant des techniques d'apprentissage par renforcement. Nous considérons deux types de pannes, et nous montrons que le modèle atteint une précision de classification de 93% dans une petite topologie. Enfin, nous appliquons la première approche basée sur l'optimisation Bayésienne pour raffiner les paramètres du réseau sur des données collectées à partir d'un réseau opérationnel. Nous extrayons les données à partir de l'interface nord du contrôleur, et nous construisons l'état du réseau selon le modèle de donnée d'entrée de GNPy. En appliquant notre processus d'apprentissage sur deux lignes de transmission, nous constatons une réduction d'erreur qui atteint 1.7~dB sur les services monitorés.

Title : Artificial Intelligence Techniques to Improve Quality of Transmission Estimation in Optical Network

Keywords: Quality of Transmission, Machine Learning, Artificial Intelligence, Parameter Optimization, Optical Networks.

Abstract: Estimating lightpath Quality of Transmission (QoT) is crucial in network design and service provisioning. Recent studies have turned to artificial intelligence (AI) techniques to improve the accuracy of QoT estimation using the data generated by the operational network. We distinguish four categories of solutions. The first category consists of building AI models to check the feasibility of a lightpath. The second category aims to predict the exact QoT performance in order to compete with analytical models. The third category uses AI to improve the performance of QoT estimation models by reducing the uncertainty on input parameters. The last category consists of improving the performance and generalization ability of AI-based solutions by retraining the models using the least amount of training samples through transfer learning techniques.

QoT models can act as part of the digital twin of the operational network by simulating the impact of new network configurations before deploying them. However, they require a perfect knowledge of the network state, consisting of a set of optical parameters that have different levels of uncertainty. Using the QoT measurements collected by the network controller, we can have a feedback about the QoT estimation inaccuracy, which can potentially be addressed using ML based techniques. In this thesis, we study this issue of uncertainty in network parameters and consider three approaches that can improve the QoT estimation in this case. We propose different learning processes in each approach, and test their performance using simulation and real data.

The first approach relies on optimizing the network parameters using the QoT estimation error as an objective function. We apply this approach through two learning processes to target QoT estimation tools based respectively on analytical model (GNPy) and Machine Learning (neural network). This approach can minimize the SNR estimation error to close to 0~dB on already trained network configuration, and reaches 0.3~dB estimation error on unseen network configurations.

In the second approach, we retrain a neural network based model to adapt it to changes in QoT due to parameters uncertainty through Transfer Learning. We show how the model can relearn the new behavior of the network without searching for the correct values of the network parameters. We can reach up to 0.5~dB in validation error with only ten new training samples.

The last approach consists of detecting parameter changes in response to failure events using reinforcement learning techniques. We consider two types of failure events. We show that the model can correctly classify the events with up to 93\% of accuracy in small network topologies.

Finally, we apply the first approach based on Bayesian Optimization algorithm to refine network parameters using data collected from a live network. We use data extracted from the north-bound interface of the network controller to build a network state based on the input data model of GNPy. Then, we apply our learning process on two transmission lines, which led to SNR estimation improvement up to 1.7~dB for the monitored services.

