Approches d'apprentissage automatique pour la prédiction de la qualité de performance dans les réseaux optiques opérationnels

par

Ameni MEZNI

MÉMOIRE PRÉSENTÉ À L'ÉCOLE DE TECHNOLOGIE SUPÉRIEURE
COMME EXIGENCE PARTIELLE À L'OBTENTION DE LA MAÎTRISE
AVEC MÉMOIRE EN GÉNIE CONCENTRATION DES RÉSEAUX DE TÉLÉCOMMUNICATIONS
M. Sc. A.

MONTRÉAL, LE 03 SEPTEMBRE 2020

ÉCOLE DE TECHNOLOGIE SUPÉRIEURE
UNIVERSITÉ DU QUÉBEC

PRÉSENTATION DU JURY

CE MÉMOIRE A ÉTÉ ÉVALUÉ

PAR UN JURY COMPOSÉ DE:

M. Christian Desrosiers, directeur de mémoire
Département de génie logiciel et TI à l'École de Technologie supérieure

Mme. Christine Tremblay, co-directrice
Département de génie électrique à l'École de technologie supérieure

M. Kim Khoa Nguyen, président du jury
Département de génie électrique à l'École de technologie supérieure

M. Aris Leivadeas, membre du jury
Département de génie logiciel et des TI à l'École de technologie supérieure

IL A FAIT L'OBJET D'UNE SOUTENANCE DEVANT JURY ET PUBLIC

LE 01 SEPTEMBRE 2020

À L'ÉCOLE DE TECHNOLOGIE SUPÉRIEURE

REMERCIEMENTS

Mes remerciements vont en premier à mon directeur et ma codirectrice de mémoire, les professeurs Christian Desrosiers et Christine Tremblay pour m'avoir offert la possibilité de mener ma recherche sur un sujet d'actualité de notre temps, un sujet qui m'a tant passionné. Ce fut un vrai plaisir de vous connaître et un honneur d'être parmi vos étudiants. J'ai beaucoup appris à votre contact. Je tiens à vous exprimer ma plus profonde reconnaissance pour le soutien, la confiance, l'intérêt que vous avez porté à mon travail ainsi qu'à votre disponibilité et vos orientations avisées. Un grand merci au professeur Mohamed Faten Zhani pour son assistance, son encouragement et les précieux conseils qu'il m'a apportés tout au long de ce projet de recherche. Je souhaite aussi remercier les collaborateurs Dave Doucet, Doug Charlton, Petar Djukic et Maurice O'Sullivan de chez Ciena pour l'aide et les remarques qui ont grandement valorisé ce travail.

Je voudrais exprimer toute ma gratitude au Ministère de l'Enseignement supérieur et de la Recherche scientifique Tunisien qui m'a accordé la bourse d'excellence en Télécommunications et qui m'a permis de mener à bien mes travaux de recherche. Je remercie aussi le Conseil de Recherches en Sciences Naturelles et en Génie (CRSNG) pour le soutien financier.

J'ai une pensée toute particulière pour mes collègues du Laboratoire des Technologies de Réseaux Dipankar, Laure, Sandra, Martine et Stéphanie avec eux j'ai passé de moments agréables et j'ai pu discuter mes travaux de recherche. Je leur exprime ma profonde sympathie et leur souhaite beaucoup de réussite.

Enfin, j'aimerais exprimer toute ma reconnaissance envers ma famille. Merci papa, de m'avoir inculqué le sens et la valeur des études. Maman, ta présence et ton appui ne tarissent jamais. À mes frères Mahmoud et Houssem qui ont toujours été présents pour moi. À mon amie Yosra, je n'oublie jamais les moments drôles qu'on passe ensemble. Merci de m'avoir soutenu pour aller de l'avant dans les moments de doute. Les mots me manquent pour exprimer ma gratitude envers mon fiancé Aymen qui a su prendre soin de notre amour. Merci d'être constamment ma source de joie et motivation.

Approches d'apprentissage automatique pour la prédiction de la qualité de performance dans les réseaux optiques opérationnels

Ameni MEZNI

RÉSUMÉ

La popularité croissante des applications de l'internet des objets, des services infonuagiques et du déploiement étendu de la 5G a produit une énorme quantité de données transportées sur Internet. De nos jours, la fibre optique constitue le système de transport de données le plus fiable et approprié pour soutenir l'ère du *Big Data*. Cependant, les opérateurs de réseaux optiques ont des défis importants pour répondre à la demande exponentielle en bande passante, de manière sécuritaire et rentable. Pour garantir une bonne qualité de transmission, une marge de sécurité statique et relativement grande est réservée lors de la conception des réseaux optiques. Cette marge prend en considération des facteurs de risque pour la transmission, tels que le vieillissement de la fibre et les fluctuations de la puissance. De ce fait, l'exploitation de l'infrastructure physique disponible est sous-optimale. Réduire la marge de sécurité à un niveau proche de zéro peut aider à maximiser la bande passante fournie. Pour ce faire, une bonne compréhension du comportement des circuits optiques s'avère indispensable. Dans ce contexte, la collecte des données de monitoring dans les réseaux optiques opérationnels pourrait être très avantageuse. En effet, elle permet de suivre quotidiennement l'évolution de la qualité de performance dans les circuits optiques. Les données de monitoring recueillies durant plusieurs saisons peuvent aussi servir dans la prédiction de la qualité de performance. Les travaux existants se sont limités à faire cette prédiction en utilisant uniquement des données synthétiques.

Ce projet étudie la prédiction de la qualité de performance traduite par le rapport signal sur bruit pour un horizon de 24 heures, et ce dans les réseaux optiques opérationnels. L'approche utilisée est celle de l'apprentissage automatique. L'étude s'est restreinte à quelques circuits optiques dont le choix est justifié. Cinq algorithmes de prédiction sont proposés et sont analysés selon les métriques de performance suivantes : le biais, l'erreur absolue moyenne, la racine carrée de l'erreur quadratique moyenne et le coefficient de détermination. Les architectures de réseaux de neurones évaluées sont : les réseaux récurrents de type *Long Short-Term Memory* (LSTM) et *Gated Recurrent Unit* (GRU), et les réseaux convolutionnels à une dimension (1D-CNN). Ces dernières sont comparées avec un modèle autorégressif intégré à moyenne mobile (ARIMA).

En conclusion, le pouvoir prédictif des modèles dépend du circuit en lui-même. Il n'est pas garanti que tous les circuits optiques opérationnels contiennent des patrons prédictibles. Certains sont prédictibles à court ou à moyen termes, d'autres sont prédictibles jusqu'à 3 jours à l'avance. Parmi les méthodes comparées, le *stateful* LSTM offre les meilleures performances. Ces résultats peuvent aussi être améliorés en appliquant l'apprentissage par transfert à partir de deux sources de données. L'ARIMA, une méthode simple donne également des prédictions satisfaisantes pour les circuits optiques non saisonniers ayant des patrons linéaires.

Mots-clés: réseau optique opérationnel, données de monitoring de performances réelles, prédiction temporelle, qualité de performance, modèle autorégressif intégré à moyenne mobile, réseaux récurrents, réseaux convolutionnels, apprentissage par transfert.

Machine Learning approaches for quality of performance prediction in operational optical networks

Ameni MEZNI

ABSTRACT

The increasing popularity of Internet of Things applications, cloud computing services and 5G mobile extensive deployment have led to a tremendous amount of data transported over the Internet. Nowadays, the optical fiber is the most reliable and appropriate data transport system to support the era of Big Data. However, optical network operators are facing significant challenges to meet the exponential demand for bandwidth in secure and cost-efficient ways. To guarantee an error free transmission, a static and relatively large safety margin is reserved when designing optical networks. This margin accounts for factors such as fiber aging and power fluctuations. As a result, the use of the available physical infrastructure is suboptimal. Squeezing the security margin to a near-zero level may help maximize the delivered bandwidth. Therefore, a rigorous understanding of the network behavior is essential. In this context, collecting performance monitoring data in operational optical networks could be very advantageous. Indeed, it makes it possible to monitor the evolution of the quality of performance in optical lightpaths on a daily basis. Real field monitoring data collected over several seasons could be used for quality of performance prediction. Existing researchs have been limited to doing such prediction using only synthetic data.

This project studies the quality of performance prediction measured by the signal to noise ratio during the next 24 hours in operational optical networks. A Machine Leaning apprach is used. The study was limited to a few optical lightpaths whose choice will be justified. Five time series prediction algorithms are proposed and evaluated according to the following performance metrics : the biais, the mean absolute error, the root mean squared error and the coefficient of determination. The neural network architectures being evaluated in here are : Long Short-Term Memory and Gated Recurrent Unit(GRU) which are recurrent neural networks and a one dimensional convolutional neural networks (1D-CNN). The architectures are compared with an Auto Regressive lntegrated Moving Average (ARIMA) model.

Results show that the effectiveness of the evaluated methods to model and predict SNR changes depends on the lightpath itself. It is not guaranteed that all operational ligthpaths contain predictable patterns. Some ligthpaths are just predictable in the short to medium term, others are predictable 3 days ahead. The stateful LSTM performs better than the other methods. Better prediction accuracy is obtained by applying transfer learing from two different data sources. The ARIMA, a simple method, also produce satisfactory prediction results for non-seasonal lightpaths with linear patterns.

Keywords: Operational optical network, real field performance monitoring data, time series prediction, quality of performance, auto regressive integrated moving average model, recurrent neural networks, convolutionnal neural networks, transfer learning.

TABLE DES MATIÈRES

Page

LISTE DES TABLEAUX

Page

LISTE DES FIGURES

Page

LISTE DES ABRÉVIATIONS, SIGLES ET ACRONYMES

Adam	Adaptive Moment Estimation
ASE	Amplified Spontaneous Emission
ASK	Amplitude Shift Keying
ADF	Augmented Dickey Fuller
ARIMA	Auto Regressive lntegrated Moving Average
ARIMAX	Auto Regressive lntegrated Moving Avenage with Exogeneous Input
ACF	Autocorrelation Function
BPTT	BacK Propagation Through Time
BPSK	Binary Phase Shift Keying
BER	Bit Error Rate
CWDM	Coarse Wavelength Division Multiplexing
CV	Coefficient of Variation
CSV	Coma Seperated Values
CRSNG	Conseil de Recherches en Sciences Naturelles et en Génie
CNN	Convolutional Neural Network
DWDM	Dense Wavelength Division Multiplexing
DGD	Differential Group Delay
DSP	Digiatl Signal Processing
DES	Double Exponential Smoothing
DP	Dual Polarisation
DP-QPSK	Dual Polarisation Quadrature Phase Shift Keying
DTW	Dynamic Time Wrapping
EDFA	Erbium Doped Fiber Amplifier

FEC	Forward Error Correction
FWM	Four-Wave Mixing
FSK	Frequency Shift Keying
FC	Fully Connected
GRU	Gated Recurrent Unit
Gbaud	Gega baud
GVD	Group Velocity Dispersion
HI	Horizontal in-Phase
HQ	Horizontal Quadrature
ITU-T	International Telecommunication Union
IQR	Interquartile Range
KNN	Interquartile Range
KPSS	Kwiatkowski-Phillips-Schmidt-Shin
LSTM	Long Short Term Memory
MLE	Maximum Likelihood Estimation
MAE	Mean Absolute Error
MMF	Multi Mode Fiber
NMS	Network Management System
NDFS	Non Dispersion Shifter Fiber
OIF	Optical Internetworking Forum
OPM	Optical Performance Monitoring
ORL	Optical Return Loss
OSNR	Optical Signal to Noise Ratio
PACF	Partial Autocorrelation Function
PSK	Phase Shift Keying

PDM	Polarisation Division Multiplexing
PBC	Polarization Beam Combiner
PBS	Polarization Beam Splitter
PMD	Polarization Mode Dispersion
QPSK	Quadrature Phase Shift Keying
QoT	Quality of Transmission
RF	Radio Fréquence
RNN	Recurrent Neural Network
ROADM	Reconfigurable Optical Add-Drop Multiplexer
RMSE	Root Mean Squared Error
SPM	Self-Phase Modulation
SPM	Self-Phase Modulation
SLA	Service Level Agreement
SNR	Signal to Noise Ratio
SMF	Single Mode Fiber
SBS	Stimulated Brillouin Scattering
SRS	Stimulated Raman Scattering
SVM	Support Vector Machine
VIF	Variance Inflation Factor
VI	Verical in-Phase
VQ	Verical Quadrature
WDM	Wavelength Division Multiplexing
WSS	Wavelength Selective Switch

LISTE DES SYMBOLES ET UNITÉS DE MESURE

dB	Décibel
dB/Km	décibel par Kilomètre
Gbit/s ou Gb/s	Gigabits par seconde
GHz	Gigahertz
Km	Kilomètre
nm	Nanomètre

INTRODUCTION

La technologie de plus en plus démocratisée et omniprésente dans le quotidien provoque une augmentation exponentielle et sans équivoque de la quantité de données disponibles. La gestion de ces données massives qualifiée de *Big Data* pose plusieurs défis notamment en matière de transport. La télécommunication optique constitue une solution potentielle et évolutive pour s'adapter aux exigences en volume et en vitesse du trafic. En 2020, la quasi-totalité des communications mondiales traversent des câbles à fibre optique.

Par contre, le contrôle des réseaux optiques est critique : toute interruption, même momentanée, peut causer d'énormes pertes des données, conduisant ainsi à une mauvaise expérience client. Le trafic du circuit défaillant doit être acheminé sur une connexion de secours, le temps que la réparation ait lieu. Le choix de cette dernière connexion n'est pas automatisé et se base essentiellement sur une expertise humaine. Pour des raisons financières, les circuits de secours ne sont pas dédiés, mais portent à leur tour un autre trafic.

Afin d'offrir à ses clients un service fiable, l'opérateur doit mettre en œuvre des mesures préventives et des mécanismes de surveillance rigoureuse sur ses réseaux. Dans une perspective de prévention, une marge de sécurité statique et relativement grande est réservée lors de la conception des réseaux optiques. Elle tient compte des pertes relatives à la dégradation des équipements, du vieillissement de la fibre, de la fluctuation de la puissance, etc. La marge de sécurité demeure inutilisée sauf dans le cas d'apparition d'un de ces facteurs ou d'emprunt de la marge pour une connexion en panne. L'exploitation de l'infrastructure physique est de ce fait sous-optimale alors que la demande en bande passante ne cesse de croitre. Les opérateurs pourraient augmenter considérablement leurs profits en se servant de cette capacité gaspillée en temps normal. Par ailleurs, ils ont eu recours au monitoring de performance optique en collectant fréquemment un certain nombre de paramètres traduisant la qualité de transmission et tout changement dans le fonctionnement des circuits optiques. Les données recueillies durant

plusieurs saisons et dans diverses conditions de déploiement constituent une source d'information précieuse, mais peu investiguée jusqu'à l'instant. Ces indicateurs ne sont consultés qu'en cas de problème intermittent affectant le réseau, soit en mode réactif.

Face à la complexité et l'hétérogénéité des réseaux, l'introduction de la cognition s'avère nécessaire. Cela permettra de bénéficier d'un réseau intelligent capable d'observer et d'analyser au mieux ses données de monitoring pour ajuster ses paramètres de transmission. Un réseau cognitif utiliserait les techniques d'apprentissage automatique pour prendre des décisions en se basant sur son état actuel et sur ses expériences passées. L'automatisation de la reconfiguration et du contrôle en général du réseau serait ainsi possible. De surcroît, l'analyse des données de monitoring permettrait de caractériser chaque connexion optique, de connaitre l'état réel de l'ensemble du réseau, de localiser rapidement les pannes, d'identifier la dégradation de la fibre et de prendre les mesures proactives nécessaires au bon moment.

La prédiction des performances des réseaux optiques à court ou à moyen terme s'avérerait très avantageuse dans ce contexte. Cette qualité de performance est couramment traduite en matière de rapport signal sur bruit électrique (*Signal to Noise Ratio* : SNR). L'opérateur serait capable d'ajuster de façon dynamique la marge de sécurité selon le besoin : il pourrait la réserver en entière partie pour le trafic ayant des accords de niveaux de services (*Service Level Agreement* : SLA) exigeants ou l'utiliser ailleurs. Il pourrait aussi automatiser le processus d'emprunt de la marge pour les connexions défaillantes.

Quelques études se sont basées sur des données de monitoring synthétiques pour prouver la possibilité de la prédiction. Des travaux récents ont commencé à utiliser des données de terrain. Les questions à la base de cette recherche sont les suivantes :

- pourrait-on prédire la qualité de performance dans des circuits optiques calmes ou dynamiques à court, à moyen ou à long termes en utilisant des données de monitoring collectées dans un réseau opérationnel ?
- y'aurait-il un modèle de prédiction capable de produire de bonnes performances quelque soit le circuit optique étudié ?

La base de données mise à disposition de cette recherche est acquise au sein des réseaux cohérents de grande capacité d'un opérateur de télécommunications. Elle contient des données de 150 circuits optiques dont certains sont enfouis et d'autres sont aériens. Le présent projet s'est limité à l'étude de quelques circuits bien particuliers, et ce pour une année de données disponible. Le choix des circuits étudiés sera justifié plus tard dans ce document. Les techniques adoptées dans ce projet dérivent de l'apprentissage automatique.

En apprentissage automatique, deux approches peuvent être adoptées pour modéliser un problème. La première est générative. Elle s'appuie sur une modélisation mathématique des probabilités conditionnelles de la distribution des données. La deuxième est discriminante. Elle cherche à maximiser l'exactitude de prédiction directement à partir des données. Cette recherche se concentre uniquement sur les modèles discriminants. La problématique peut être formulée comme étant une prédiction appartenant à la famille d'apprentissage supervisé.

Une grande panoplie d'algorithmes existe en apprentissage automatique y compris les réseaux de neurones. Inspirés du fonctionnement du cerveau humain, les réseaux de neurones sont capables de résoudre des tâches aussi complexes surtout en présence d'une quantité importante de données. Ces modèles sont largement utilisés dans la reconnaissance d'images, le traitement des paroles, mais aussi dans la prédiction des séries temporelles. Il existe deux catégories de réseaux de neurones. Le premier est le réseau de neurones récurrent à savoir le *Long Short Term Memory* (LSTM) et le *Gated Recurrent Unit* (GRU). Le deuxième est le réseau de neurones à propagation avant tel que le réseau convolutionnel (CNN). Le LSTM, GRU et CNN ont connu un succès

retentissant dans la prédiction des données séquentielles. Ils sont ainsi proposés dans notre cas pour la prédiction de la qualité de performance. Ces méthodes sont étudiées et comparées à une baseline de type ARIMA (*Auto Regressive lntegrated Moving Avenage*). De nombreuses métriques de performance ont été utilisées dans cette comparaison telles que la racine carrée de l'erreur quadratique moyenne, l'erreur absolue moyenne, le coefficient de détermination, etc. Dans le même contexte, plusieurs variables explicatives peuvent être introduites à l'entrée des réseaux de neurones en vue d'améliorer la qualité de prédiction. Une étude a été faite dans ce document pour les identifier.

En absence de quantité suffisante de données, la communauté scientifique a recours à l'utilisation de l'apprentissage par transfert avec les techniques d'apprentissage automatique. En apprentissage par transfert, un modèle entraîné à faire une tâche est réutilisé comme initialisation d'un autre entraînement pour réaliser une deuxième tâche connexe. L'effet de l'apprentissage par transfert à partir de deux sources de données sur la qualité de la prédiction est aussi investigué.

La prévisibilité de la qualité de performance traduite par le rapport signal sur bruit du récepteur (SNR) est prise comme hypothèse principale de cette recherche. En d'autres termes, il ne s'agit pas d'un phénomène purement aléatoire.

Les objectifs de cette recherche sont les suivants :

- développer un algorithme à base de réseau de neurones pour prédire la qualité de performance (traduite par le SNR) en utilisant les données de monitoring collectées dans des conditions de terrain réelles ;
- étudier les attributs explicatifs qui pourraient influencer la variable prédite SNR ;
- investiguer l'effet de l'apprentissage par transfert à partir de deux sources de données sur la prédiction de la qualité de performance.

Le rapport est structuré comme suit :

Chapitre 1 Passe en revue les connaissances théoriques sur le monitoring des réseaux optiques cohérents directement reliées avec ce projet.

Chapitre 2 La première partie de ce chapitre revoit les connaissances de base sur les séries temporelles. Les principes de la baseline ARIMA, les réseaux de neurones évalués dans ce projet ainsi que l'apprentissage par transfert y sont aussi présentés. La dernière partie discute des travaux déjà menés dans ce domaine ainsi que leurs limites.

Chapitre 3 Expose la méthodologie adoptée, commençant par une analyse descriptive et exploratoire des circuits sélectionnés. Viennent ensuite le traitement et la préparation des données pour les algorithmes d'apprentissage automatique. Le chapitre explique aussi la méthode de sélection des attributs explicatifs qui peuvent être intégrés aux entrées des réseaux de neurones. De même, il présente les critères de performance utilisés dans l'évaluation des modèles prédictifs. Par la suite, les stratégies d'entraînement et de l'application de l'apprentissage par transfert pour prédire la qualité de performance sont fournies.

Chapitre 4 Présente une analyse comparative des différents résultats obtenus avec les modèles prédictifs. Plusieurs scénarios sont ensuite evalués.

Chapitre 5 Conclut ce document et discute des perspectives d'amélioration.

L'application et la comparaison du LSTM, GRU et 1D-CNN et l'utilisation de l'apprentissage par transfert pour la prédiction de la qualité de performance dans les réseaux optiques opérationnels sont nouvelles dans le domaine de la fibre optique. Une partie de cette étude a fait l'objet d'une publication (Mezni, Charlton, Tremblay & Desrosiers, 2020) dans le cadre de la conférence *'Conference on Laser and Electro-Optics'* (CLEO). Cette dernière est un événement mondial qui

rassemble des leaders de la recherche scientifique et des professionnels dans le domaine de la fibre optique.

CHAPITRE 1

REVUE DE CONNAISSANCE SUR LE MONITORING DES RÉSEAUX OPTIQUES COHÉRENTS

1.1 Introduction

L'objectif général de ce projet est d'étudier la prédiction de la qualité de performance dans les réseaux optiques opérationnels cohérents. L'approche utilisée se base sur les techniques d'apprentissage automatique appliquées sur les données de monitoring de performance. Ce chapitre est consacré aux notions relatives au monitoring des performances des réseaux optiques cohérents à grande capacité. À cet égard, les concepts de base de la communication optique et ses avantages sont introduits. Ensuite, l'architecture d'une liaison à multiplexage en longueur d'onde ainsi que les phénomènes physiques de dégradation affectant les réseaux à fibre optique sont présentés. Par après, les critères de performances dans les réseaux optiques sont couverts. L'évolution vers les réseaux cohérents à grande capacité ainsi que leurs particularités sont aussi expliquées. Finalement, les objectifs du monitoring des réseaux optiques sont décrits.

1.2 Concepts fondamentaux de la communication optique

Tous systèmes capables d'échanger un signal entre un émetteur et un récepteur à travers un canal tout en garantissant une certaine qualité de transmission sont appelés systèmes de communication. L'architecture de base d'un tel système est illustrée dans la figure 1.1. Le signal subit plusieurs transformations avant d'être acheminé dans le canal, commençant par l'échantillonnage et la quantification jusqu'à l'encodage et la modulation. Ces étapes sont représentées par le traitement du signal dans la figure ci-dessous. Dégradé par l'effet de son passage dans un canal bruité, le signal sera traité pour pouvoir récupérer l'information utile au niveau du récepteur. Dans le contexte des communications optiques, un autre élément clé vient s'ajouter à l'architecture précédente : c'est le transducteur. Ce dispositif fait correspondre la propriété du signal à celle du canal via traitement du signal. Ainsi, le signal électronique venant de l'émetteur est converti en signal optique correspondant par le transducteur électrooptique (E-O), illustré dans la figure 1.2.

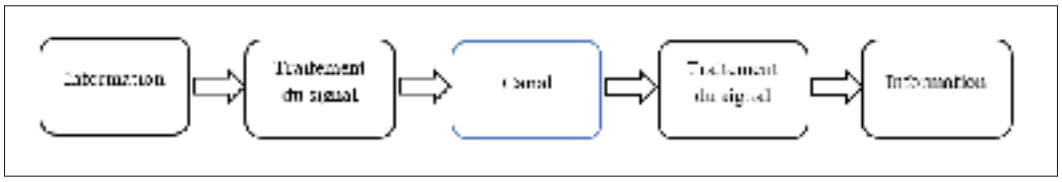


Figure 1.1 Architecture de base d'un sysème de communication
Adaptée de Tremblay (2018)

De plus, le signal optique ayant propagé dans le canal de transmission sera reconverti en signal électronique par le transducteur optoélectronique (O-E) avant d'être reçu en destination. Le laser et le photodétecteur sont des exemples typiques de dispositifs E-O et O-E respectivement. Le canal de communication utilisé dans le cadre de ce projet est la fibre optique. Il existe deux grandes catégories de fibre optique à savoir le monomode (*Single Mode Fiber* : SMF). La fibre monomode possède un diamètre du coeur de quelques microns, permettant la propagation d'un seul mode. Cependant, la fibre multimode (*Multi Mode Fiber* : MMF) laisse passer plusieurs rayons lumineux (autrement dit longueurs d'onde) qui suivent des chemins différents. Ils peuvent donc arriver en destination à des moments différents induisant une certaine dispersion du signal optique. L'utilisation du MMF se limite aux courtes distances. Dans ce projet, la fibre monomode est celle utilisée. La fibre optique offre plusieurs avantages dont certains seront détaillés dans la section suivante.

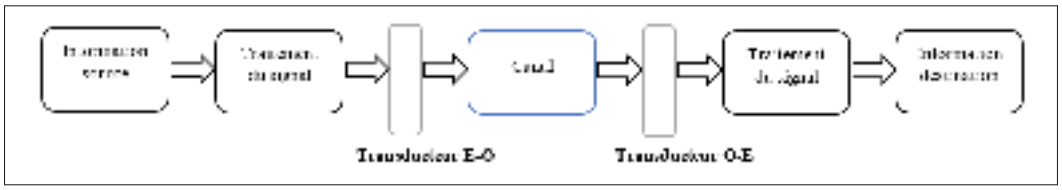


Figure 1.2 Architecture de base d'un système de communication optique
Adaptée de Tremblay (2018)

1.2.1 Avantage de la fibre optique

Trois facteurs ont toujours motivé le développement de chaque nouveau système de communication (Keiser, 2010). Premièrement, l'amélioration de la qualité du signal reçu est visée. Dans ce cas, moins d'erreurs et moins de distorsions dans l'information acheminée sont obtenus. Deuxièmement, il est nécessaire d'étendre davantage la capacité de transmission et donc de transporter plus d'information en utilisant la même infrastructure physique. Troisièmement, il est préférable d'augmenter la distance de transmission entre deux régénérateurs de signal ou encore deux stations d'amplifications.

La fibre optique a fait preuve de performance dans les trois derniers aspects mentionnés et même plus. L'ère de la fibre optique a débuté avec un article scientifique sur la théorie et les applications des communications par fibre optique publiée en 1966 (Kao & Hockham, 1966). Il est venu révolutionner le monde des télécommunications optiques. Mettre l'accent sur le développement des systèmes de transmission par fibre optique est devenu un besoin urgent et indispensable surtout devant la prolifération exponentielle de la quantité de données disponible depuis les années 1990. La fibre optique présente plusieurs avantages par rapport au fil de cuivre largement déployé dans les systèmes de communication électrique auparavant (Keiser, 2010). D'abord, elle présente une faible perte comparée au cuivre : les données peuvent être envoyées sur une plus grande distance. Ainsi, l'ajout d'équipements de restauration sera moins fréquent. La fibre a aussi une plus grande capacité que le cuivre : l'opérateur pourra mettre en place moins de liaison optique que de liaison en fil de cuivre pour transmettre la même quantité de données. De surcroit, la fibre optique est un matériau diélectrique non conductible de l'électricité. Ceci lui donne une immunité contre l'interférence électrique et lui garantit une meilleure qualité de transmission, un avantage inexistant avec le cuivre. En termes de sécurité de l'information, la fibre est plus sécuritaire étant donné que le signal est confiné à l'intérieur d'une gaine et d'un revêtement protecteur : le signal est difficilement intercepté comparé au fil de cuivre. Finalement, la fibre a de faible poids et dimension par rapport au cuivre ce qui lui offre un large domaine d'utilisation.

Avant d'être acheminé en destination, le signal doit être modulé selon les propriétés du canal de transmission. La modulation, concept fondamental dans les réseaux de communication, sera expliqué dans la section qui suit.

1.2.2 Formats de modulation

Dans les systèmes de télécommunications, il est nécessaire de représenter le signal portant l'information sur une onde porteuse dont la fréquence est adaptée au canal de transmission. La modulation numérique consiste à contrôler l'amplitude, la fréquence ou la phase de l'onde porteuse en fonction du message à transmettre. Il existe donc trois grandes catégories de modulation selon le paramètre varié : modulation d'amplitude (*Amplitude Shift Keying* : ASK), modulation de fréquence (*Frequency Shift Keying* : FSK) et modulation de phase (*Phase Shift Keying* : PSK) impliquant tous un nombre discret de symboles permettant la transmission d'informations. Ces symboles caractérisés par une amplitude et une phase, peuvent être représentés dans un plan complexe appelé aussi plan I/Q (*In phase* : I ou encore en phase, *Quadrature* : Q ou encore en quadrature de phase). Cette représentation est connue sous le nom de diagramme de constellation et les symboles forment les points de constellation. Une modulation ayant un plus grand nombre de points de constellation permet d'acheminer plus d'informations par symbole. En communication optique, le format de modulation le plus commun est le PSK (Karlsson & Agrell, 2009)

$$x(t) = A_c sin[2\pi f_c + \theta(t)] \tag{1.1}$$

Où :

A_c : amplitude de l'onde porteuse (valeur constante),

f_c : fréquence de l'onde porteuse,

$\theta(t)$: phase du signal modulé. Elle varie selon le message à transmettre.

Désignons les formats de modulation de phase par XPSK où X représente le nombre de points de constellation possibles. Par exemple, BPSK peut transmettre un bit par symbole (0 ou 1), soit deux points de constellation, où la phase du signal modulé prend les valeurs 0 ou π selon la valeur du bit. Également QPSK (*Quadrature Phase Shift Keying*) peut transmettre deux bits

par symbole (00, 01, 10 ou 11), soit quatre points de constellation en tout. La phase du signal modulé peut prendre une valeur parmi : $\frac{-3\pi}{4}, \frac{-\pi}{4}, \frac{\pi}{4}, \frac{-3\pi}{4}$.

En développant l'équation 1.1 tout en fixant la phase avec une des valeurs précédemment mentionnées, elle devient :

$$x(t) = A_1 cos(2\pi f_c) + A_2 sin(2\pi f_c) \tag{1.2}$$

Où :

A_1 : amplitude de la composante cosinus,

A_2 : amplitude de la composante sinus.

De ce fait, le signal modulé peut être décomposé en la somme de deux signaux, l'un en phase et l'autre en quadrature de phase. Ces derniers sont orthogonaux et n'interfèrent pas ensemble.

Dans le but d'augmenter davantage la capacité du canal, l'information est envoyée sur les deux polarisations du signal, horizontale et verticale, en même temps. Cette technique est souvent appelée multiplexage de polarisation (*Polarisation Division Multiplexing* : PDM) ou encore (*Dual polarisation* : DP). La combinaison de la modulation QPSK avec le DP (nommé DP-QPSK), permet de doubler le débit de transmission. Prenons l'exemple d'un canal de 100 Gbit/s, uniquement entre 25 et 28 Gbit/s sont transmise : chaque symbole contient quatre bits. Pour ce débit, l'OIF (*Optical Internetworking Forum*) recommande déjà l'utilisation du DP-QPSK. Sa performance a été démontrée par Nortel (ancien nom de Ciena) depuis 2008 (Lecoy, 2015). Le DP-QPSK est le format de modulation utilisé dans le cadre de ce projet. Les formats de modulation posent un compromis entre l'efficacité spectrale et la puissance du signal de chaque bit. Ainsi, en utilisant une modulation plus élevée, un nombre de symboles plus large sont transmis par seconde au détriment de la puissance du signal et par conséquent la distance parcourue . Les réseaux à Multiplexage en longueur d'onde sont le type de réseau dominant aujourd'hui. La section suivante explique comment un tel réseau fonctionne et illustre son architecture de base et introduit ses sous-types.

1.3 Liaison optique WDM à longue distance

Les liaisons par fibre optique longue distance à très grande capacité sont fréquemment utilisées. L'architecture la plus commune de ce type de liaison est schématisée dans la figure 1.3. En général, une liaison est dite longue distance si elle parcourt quelques centaines voire des milliers de kilomètres entre la source et la destination. Pour répondre à la demande croissante en bande passante, les réseaux WDM ont été développés. WDM est l'abréviation de *wavelength division multiplexing* ou encore le multiplexage en longueur d'onde. C'est une technologie qui permet d'injecter simultanément sur la même fibre plusieurs canaux optiques indépendants dont chacun correspond à une longueur d'onde donnée (dite aussi couleur). Une longueur d'onde est définie par sa position dans la grille de fréquence ainsi que sa largeur spectrale. Ces deux paramètres sont donnés soit en nanomètres (nm), soit en gigahertz (GHz). L'espacement entre les canaux optiques est normalisé par l'organisme mondial ITU-T (*International Telecommunication Union* : ITU-T). En émission, N signaux sont placés sur N longueurs d'onde différentes grâce à N transpondeurs. Ensuite, le multiplexeur noté Mux convertit ces N signaux en un seul puis l'envoie sur la fibre optique. En réception, l'opération inverse est assurée par le démultiplexeur noté Démux. Il permet de séparer les N différentes longueurs d'ondes chacune à part et les envois aux transpondeurs O-E.

Il existe deux catégories dans la technologie WDM à savoir CWDM (*Coarse* WDM) et DWDM (*Dense* WDM). Alors qu'un réseau CWDM peut avoir jusqu'aux 18 canaux optiques multiplexés dans la bande située entre 1271 nm et 1611 nm et dont chacun est espacé de 20 nm, un réseau DWDM peut contenir jusqu'aux 80 canaux aux alentours de la bande 1550nm, séparés chacun de 0.8 nm. À cause du large espacement entre les canaux pour les réseaux CWDM, les opérateurs peuvent utiliser des lasers non coûteux. Cependant, aucune amplification n'est possible dans leurs bandes de transmission ce qui rend leurs utilisations limitées aux courtes distances. Les réseaux DWDM sont plutôt adaptés aux transmissions longues distances puisque l'amplification est faisable dans leurs bandes. Les modules d'ajout/extraction de canaux optiques reconfigurables (*Reconfigurable Optical Add-Drop Multiplexer* : ROADM) à commutateur sélectif en longueur d'onde (*Wavelength Selective Switch* :WSS) sont largement déployés dans les réseaux WDM à

longue distance. Ils permettent d'ajouter et/ ou extraire n'importe quelles longueurs d'onde à n'importe quel point du réseau tout en laissant passer les autres. Les longueurs d'ondes extraites peuvent être réutilisées de nouveau dans la liaison optique. Ce module fonctionne sans avoir recours à une conversion électronique du signal. Le WSS, composante principale dans un ROADM, utilise la technique de prisme et miroirs pour séparer ou sélectionner les couleurs voulues. Le ROADM offre plus de flexibilité et évolutivité aux réseaux optiques : il permet d'optimiser l'exploitation de la fibre existante en injectant et/ou extrayant des canaux dans des sites différents. Pour pallier l'effet de l'atténuation du signal optique, des amplificateurs dopés à l'erbium (*Erbium Doped Fiber Amplifier* : EDFA) sont ajoutés aux liaisons à longue distance. La fonction de l'amplificateur diffère selon son emplacement dans la liaison. Trois catégories d'amplificateurs optiques sont illustrées dans l'architecture ci-dessous :

- amplificateur de puissance : placé juste en sortie de l'émetteur. Il permet d'amplifier le signal transmis au début de la liaison ;
- préamplificateur : placé en entrée du récepteur. Il permet d'amplifier le signal reçu et par conséquent améliorer le rapport signal sur bruit (*Signal to Noise Ratio* : SNR) ;
- amplificateur de ligne : permet de compenser exactement la perte de signal encourue après avoir traversé une section de fibre de longueur l (appelé aussi *span*). L'amplification se fait sur le signal multiplexé.

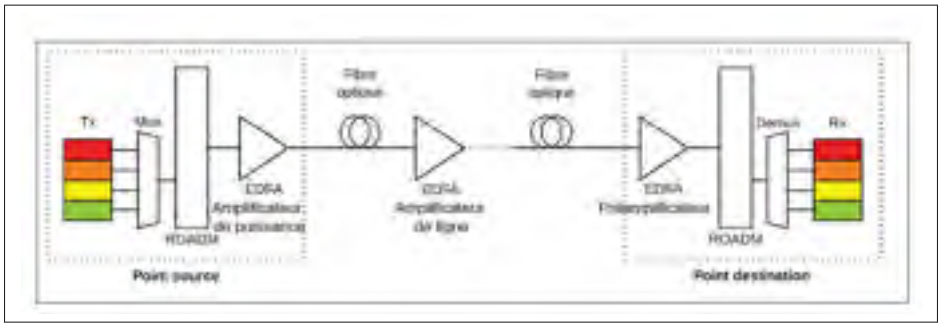


Figure 1.3 Schéma d'une liaison optique WDM à longue distance

Durant sa propagation entre la source et la destination, le signal subit plusieurs dégradations dont certains sont linéaires et d'autres sont non linéaires. Ces phénomènes de dégradation, discutés dans la section suivante, altèrent la qualité du signal et limitent la distance de transmission.

1.4 Phénomènes physiques de dégradation dans le circuit optique

En communication optique, les termes linéaire et non linéaire réfèrent respectivement à des phénomènes dépendants ou non de la puissance du signal lumineux. Dans les réseaux de transmission synchrone traditionnels, les effets de dégradation se manifestant dans la fibre sont linéaires puisque la puissance optique du signal incident est faible. De ce fait, les principaux phénomènes de dégradation affectant la fibre sont l'atténuation et la dispersion. Cependant, dans les réseaux WDM, une dizaine voire une centaine de longueurs d'onde périodiquement amplifiées se propagent au sein de la même fibre optique. Un couplage entre les différents signaux optiques caractérisés par leurs puissances élevées se produit, favorisant ainsi l'apparition des effets non linéaires. Les effets de dégradation non linéaire constituent un facteur clé limitant les performances des réseaux WDM.

1.4.1 Effets linéaires de dégradation dans le circuit optique

La propagation de la lumière dans une fibre optique est influencée par de nombreux phénomènes physiques linéaires y compris l'atténuation et la dispersion chromatique qui sont présentés dans la figure 1.4. Ces derniers sont des caractéristiques pour chaque type de fibre et affectent considérablement la transmission du signal optique dans les réseaux opérant à 10 Gb/s et plus. Dans ce qui suit, ces deux phénomènes sont expliqués tout en indiquant leurs provenances ainsi que leurs effets sur la propagation du signal.

1.4.1.1 L'atténuation dans la fibre optique

L'atténuation représente l'affaiblissement du signal optique au fur et à mesure de sa propagation dans la fibre. Comme le montre l'équation 1.3, la puissance optique décroit de façon exponentielle

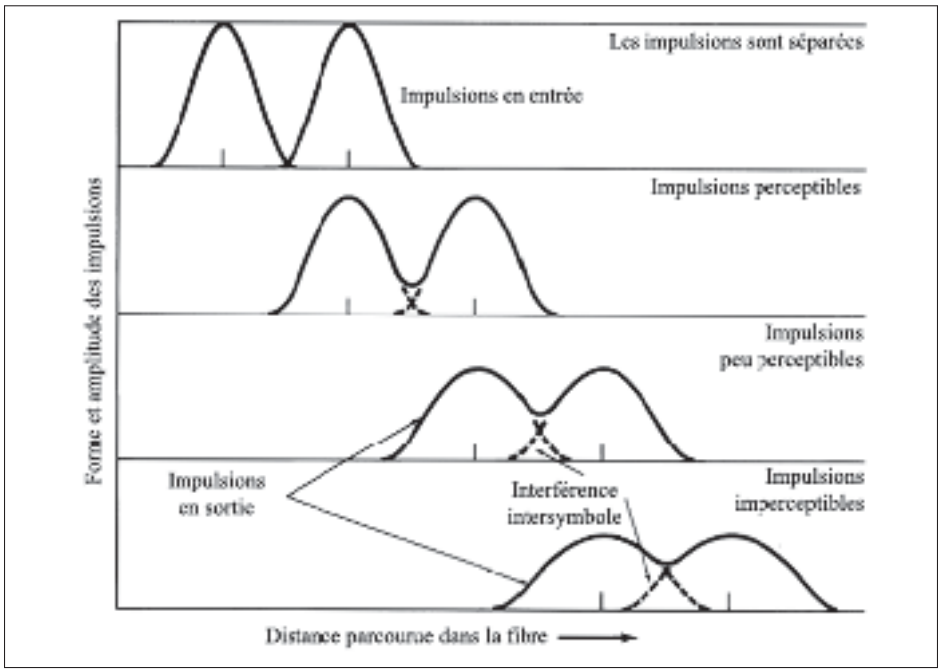


Figure 1.4 Effets des phénomènes de l'atténuation et de la dispersion sur le signal optique
Adaptée de Keiser (2010)

en fonction de la distance parcourue, engendrant ainsi une baisse de l'amplitude du signal. Le récepteur aura une difficulté à distinguer la puissance d'un bit '1' et '0'. De ce fait, l'atténuation limite la distance de propagation et contribue à la détérioration de la qualité de transmission. Dans l'équation, $P(0)$ correspond à la puissance au point d'origine ($z = 0$) tandis que $P(z)$ constitue la puissance à une distance z.

$$P(z) = P(0)\exp^{-\alpha_p z} \tag{1.3}$$

Où :

α_p : désigne le coefficient d'atténuation de la fibre exprimée en unité de distance (à savoir Km^{-1}).

L'UIT, dans sa version UIT-R V.574-5 recommande l'utilisation du décibel (dB) pour exprimer la relation entre deux grandeurs de champs, à savoir la puissance. Couramment, le coefficient d'atténuation est traduit en (dB/km).

Ce coefficient est un facteur important pour déterminer la distance maximale sans amplification séparant l'émetteur du récepteur. Il varie généralement avec la longueur d'onde qui traverse la fibre. À titre d'exemple, le coefficient d'atténuation est à 0.22 dB/km dans la bande C ($1530 \leq \lambda \leq 1560nm$).

L'affaiblissement de la fibre est causé principalement par les phénomènes physiques suivants : l'absorption, la diffusion et les pertes par radiation de l'énergie optique. L'absorption est due à l'impureté du matériel constituant la fibre. Quant à la diffusion, elle est reliée à plusieurs facteurs y compris les variations microscopiques de la densité du matériel, l'inhomogénéité et les défauts durant la construction de la fibre, etc. Finalement, les perturbations microscopiques et macroscopiques de la géométrie de la fibre engendrent les pertes par radiation. D'autres atténuations sont également générées par le câblage de la fibre. Pour pallier l'effet de l'atténuation, on peut avoir recours à l'amplification ou moins fréquemment la régénération du signal optique chaque fois que la distance minimale entre la limite en atténuation et la limite en dispersion est parcourue.

1.4.1.2 La dispersion chromatique dans la fibre optique

De façon générale, la dispersion correspond à l'étalement temporel d'une impulsion lumineuse qui se propage dans la fibre optique. Ce phénomène limite le débit de transmission. En tout, il existe trois grandes catégories de dispersion. Premièrement, la dispersion intermodale qui apparait uniquement dans la fibre multimode lorsque ses différents rayons lumineux se propagent à des vitesses différentes. Cette dispersion ne sera pas détaillée étant donné que la fibre monomode est utilisée dans ce projet. Deuxièmement, la dispersion chromatique qui représente le sujet d'intérêt dans cette section. Troisièmement, la dispersion des modes de polarisation qui sera détaillée dans la prochaine section.

En fait, pour une distance donnée, chaque longueur d'onde se propage dans la fibre optique à une vitesse différente entraînant ainsi l'élargissement des impulsions lumineuses (Keiser, 2010). Ceci explique bien la deuxième appellation de la dispersion chromatique : dispersion de vitesse de groupe (*Group Velocity Dispersion* : GVD). Plusieurs facteurs influencent cette variation de vitesse en fonction de la longueur d'onde y compris le changement des indices de réfraction en rapport aux longueurs d'onde dans la fibre, le type du matériau formant la fibre, etc. Chaque type de fibre optique possède son propre coefficient de dispersion chromatique qui est fourni par son fabricant.

1.4.1.3 Délai de Groupe Différentiel et dispersion modale de polarisation dans la fibre optique

Une propriété fondamentale d'un signal optique est la polarisation. Cette dernière fait référence à l'orientation du champ électrique du signal qui peut varier considérablement le long de la fibre. À une longueur d'onde donnée, l'énergie du signal occupe deux modes de polarisation orthogonaux. Étant soumise à des contraintes thermiques, mécaniques, de fabrication ou aussi d'installation sévères, la symétrie circulaire du cœur de la fibre optique est compromise et des sections ayant de légères biréfringences sont créées. Par conséquent, les deux directions de propagation X et Y de la lumière auront des indices de réfraction légèrement différents ce qui a une influence directe sur la vitesse de propagation selon ces axes : les deux modes de polarisations d'une fibre monomode s'y propagent ainsi à des vitesses différentes. Il s'agit ici d'une forme de dispersion intramodale appelée délai de groupe différentiel (*Differential Group Delay* : DGD). Le phénomène de DGD entraine la distorsion des impulsions lumineuses et la dégradation en générale du système limitant ainsi la capacité de transmission de la fibre optique. Contrairement à la dispersion chromatique, le DGD est un effet linéaire dépendant du temps ce qui rend sa compensation difficile. De plus, le changement aléatoire de la biréfringence tout au long de la fibre résulte en un transfert aléatoire d'énergie entre les deux modes de propagation de la lumière. Quant à la dispersion modale de polarisation (*Polarization Mode Dispersion* : PMD) est directement liée au DGD. Elle correspond à la valeur moyenne du DGD. Son intensité varie de façon proportionnelle à la racine carrée de la longueur de fibre optique.

1.4.2 Effets non linéaires de dégradation dans le circuit optique

Les effets de dégradation non linéaires sont des phénomènes physiques très remarquables en communication optique qui surviennent en résultat de l'accumulation de certains facteurs déclenchants et lorsque la puissance du signal lumineux dépasse une certaine valeur seuil. Généralement, l'opérateur est responsable de repérer les seuils de puissance au-dessus desquels les effets non linéaires apparaissent. Il ajuste ensuite son système de transmission pour fonctionner sous le seuil de l'effet non linéaire le plus contraignant. Ces effets sont principalement causés par les processus non linéaires de diffusion inélastique et par les variations non linéaires de l'indice de réfraction de la fibre optique, appelé aussi effet de Kerr.

1.4.2.1 Processus non linéaires de diffusion inélastique

Les processus non linéaires de diffusion inélastique se divisent en diffusion Raman Stimulée et diffusion Brouillon Stimulée. Ils sont issus tous les deux de l'interaction entre les photons du signal optique et les modes vibrationnels des molécules de silice formant la fibre.

Diffusion Raman Stimulée (*Stimulated Raman Scattering* : SRS) :

Dans un réseau WDM, le SRS se manifeste par un transfert d'énergie des longueurs d'onde les plus basses aux longueurs d'onde les plus élevées. En d'autres termes, il y a diffusion de la lumière de l'onde incidente à une longueur d'onde plus élevée entrainée par l'absorption d'une portion de l'énergie du photon incident dans la silice (Singh & Singh, 2007).

Diffusion Brouillon Stimulée (*Stimulated Brouillon Scattering* : SBS) :

La diffusion Brouillon stimulée correspond à un phénomène de rétrodiffusion de lumière originaire d'une interaction paramétrique entre une onde de pompe, une onde Stockes et finalement une onde acoustique dans un canal optique. La lumière diffusée subit un décalage Doppler en fréquence. Généralement, l'opérateur installe des isolateurs optiques sur les sites d'amplification de lignes. Ceci permet de bloquer la propagation du signal SBS diffusé. De ce fait, le SBS est un effet intracanal par span.

1.4.2.2 Effet de Kerr

Étant dépendant de la puissance du signal optique, l'indice de réfraction est le premier responsable de l'effet de Kerr. Selon du type de signal optique incident, l'effet de Kerr peut se manifester sous trois formes (Singh & Singh, 2007) :

- mélange à quatre ondes (*Four-Wave Mixing* : FWM) ;
- automodulation de phase (*Self-Phase Modulation* : SPM) ;
- modulation de phase croisée (*Cross-Phase Modulation* : XPM).

1.5 Comment mesurer la qualité du signal optique ?

Mesurer la qualité du signal optique est une tâche très importante en communication optique et requiert une attention particulière. Plusieurs métriques sont couramment utilisées à savoir le rapport signal sur bruit optique, le taux d'erreur sur les bits (*Bit Error Rate* : BER), le facteur Q, le rapport signal sur Bruit (SNR), la puissance du signal optique, etc. Une corrélation existe entre certains différents paramètres. Le BER s'avère la métrique de qualité la plus concluante (Freude, Schmogrow, Nebendahl, Winter, Josten, Hillerkuss, Koenig, Meyer, Dreschmann, Huebner et al., 2012). Chacun de ces paramètres sera expliqué dans ce qui suit.

1.5.1 Taux d'erreurs sur les bits et ses variantes

Taux d'erreur sur les bits :

Le BER est une mesure du nombre de bits en erreurs parmi le nombre total de bits transmis pendant une période temporelle donnée. Ces erreurs peuvent apparaitre au niveau du récepteur lors du processus de prise de décision par rapport à la valeur du bit reçu. Le BER est généralement exprimé sous forme de rapport sans unité. Par exemple, s'il y a trois bits erronés parmi un total de 1 million de bits transmis, le taux d'erreur sur les bits sera de $BER = \frac{3}{1000000} = 3 \times 10^{-6}$. Cette métrique traduit la qualité de l'émission, la réception et le canal de transmission dans lequel s'est propagé le signal ainsi que l'environnement affectant la fibre optique. En fait, le

BER prend en considération des facteurs tels que le bruit, l'atténuation, etc. (Frenzel, 2016). L'opérateur est responsable de maintenir un taux d'erreur sur les bits seuil ou requis (BER_{requis}) au-dessous duquel le système subira des pertes importantes. Dans ce présent projet, un réseau optique cohérent modulé DP-QPSK et ayant un débit de 100 Gbit/s requiert un BER minimal de l'ordre de 10^{-3}. Si le BER mesuré est proche du BER_{requis} , il s'avère nécessaire de diminuer le débit de données utilisé (Alam, Alam, Hu & Mehrab, 2011). La mesure du BER se fait au niveau du récepteur avant d'appliquer le code correcteur d'erreurs (*Forward Error Correction* : FEC), appelé Pre-FEC BER. L'opérateur pourra également fournir une estimation du BER après l'application du code correcteur d'erreurs et sera dans ce cas nommé Post-FEC BER. Le FEC représente une technique de correction d'erreurs qui permet de détecter et ensuite corriger un nombre limité d'erreurs dans les données transmises sans avoir besoin à la retransmission des bits erronés. Ceci se fait en ajoutant des informations supplémentaires utiles à la détection et correction d'erreurs en émission. (Keiser, 2010) rapporte que le FEC peut diminuer la valeur de BER mesuré par un facteur de 10 ce qui est favorable en termes de qualité et coût de transmission. La mesure du BER requiert l'utilisation d'un système de recouvrement d'horloge et de données ce qui est coûteux surtout pour les systèmes de communications optiques ayant une modulation complexe.

Facteur Q :

Le facteur Q représente à son tour un paramètre clé pour évaluer la qualité de performance d'un canal de communication. Il mesure le niveau de bruit dans une impulsion lumineuse pour des fins de diagnostic. Il est fortement relié au rapport signal sur bruit électrique (SNR) calculé en bout de liaison au niveau du récepteur. Pour un signal donné, le facteur Q suggère le SNR minimum requis pour maintenir une valeur de BER spécifique.

Généralement, il est possible de déterminer la valeur de cette métrique à partir du diagramme d'œil du signal. Également, pour tout type de réseau, le facteur Q peut être calculé numériquement en connaissant la valeur de BER selon la formulation suivante :

$$Q = \sqrt{2} \times \text{erfcinv}(2 \times \text{Pre-FEC BER}) \tag{1.4}$$

Ainsi, de petites valeurs de Pre-FEC BER implique de large facteur Q et de meilleures performances de liaisons. Cette métrique reflète les effets de dégradation linéaire (atténuation et dispersion chromatique) et non linéaire (dispersion des modes de polarisation) de la fibre.

Rapport signal sur bruit :

Le rapport signal sur bruit électrique (*Signal to Noise Ratio* : SNR) permet de quantifier l'impact du bruit sur la qualité du signal. Il est défini comme étant le rapport entre la puissance du signal par la puissance du bruit. Le SNR est généralement exprimé en dB par :

$$SNR(dB) \;=\; 10\log_{10}\left(\frac{P_{\text{signal}}}{\text{P}_{\text{bruit}}}\right) \tag{1.5}$$

Dans le cadre de ce projet, le SNR est calculé à partir du Pre-FEC BER selon une formulation fournie par le partenaire industriel propre à ses équipements et au format de modulation utilisé (DP-QPSK). Le SNR est choisi dans notre cas comme quantificateur de la qualité de performance étant donnée que c'est la métrique la plus interprétable dans le domaine de la fibre optique.

1.5.2 Rapport signal sur bruit optique

Le rapport signal sur bruit optique (*Optical Signal to Noise Ratio* : OSNR) est un autre indicateur de l'état du réseau optique. Cette métrique facilite aussi le diagnostic des pannes. Il est défini comme étant le rapport entre la puissance du signal optique et la puissance de bruit optique. La source prédominante de bruit optique provient de l'utilisation d'une cascade d'amplificateurs EDFA. En fait, chaque amplificateur introduit une quantité de bruit dans le signal optique dû au phénomène d'émission spontanée amplifiée (*Amplified Spontaneous Emission* : ASE). Ces petites quantités s'accumulent conduisant à la détérioration de la qualité du signal optique au bout de la liaison. L'OSNR d'un système de communication optique doit être maintenu au-dessus d'un OSNR seuil ($\text{OSNR}_{\text{requis}}$). Une forte corrélation existe entre l'OSNR et le BER. En effet, Une valeur d'OSNR élevée implique un BER faible et par conséquent un système performant.

1.5.3 Puissance du signal

La puissance du signal est un autre paramètre physique qui est mesurée à de divers emplacements dans le réseau pour suivre l'évolution de l'atténuation dans le signal. Généralement la puissance est enregistrée à la sortie de l'émetteur, avant et après les amplificateurs et à l'entrée du récepteur. L'unité la plus commune pour cette métrique est le décibel (dB).

1.6 Évolution vers les réseaux optiques cohérents à longue distance

1.6.1 La technologie cohérente et ses particularités

Les réseaux WDM permettent de multiplexer plusieurs longueurs d'onde et les envoyer simultanément sur la même fibre optique. Cette technologie a permis d'augmenter le débit de transmission considérablement à l'époque. Les opérateurs ont continué d'ajouter des canaux optiques jusqu'à atteindre la limite de l'électronique utilisée. Face à la demande continuelle en bande passante, il fallait développer une nouvelle technologie permettant plus de débit en un moindre coût et tout en exploitant l'infrastructure existante : la technologie cohérente a été déployée en combination avec des formats de modulation plus sophistiqués et un traitement de signal assez avancé (*Digiatl Signal Processing* : DSP). De ce fait, les opérateurs ont pu mettre en place des réseaux WDM avec des canaux à 100 Gbit/s et plus chacune. La technologie cohérente utilise des modulations en amplitude et en phase sur les deux polarisations horizontale et verticale de la lumière pour transmettre une plus grande quantité de données sur la même fibre. Ce projet se limite à la modulation DP-QPSK dans le cadre ce projet dont le principe a été discuté dans la section 1.2.2. La technologie cohérente implique à la fois l'utilisation d'une détection cohérente et un module de traitement de signal DSP au récepteur, deux notions qui seront expliquées dans ce qui suit.

Détection cohérente :
La détection directe est la méthode traditionnellement utilisée dans les récepteurs conventionnels. Le principe est simple : le signal ayant propagé dans la fibre est détecté par une photodiode

dont la sortie est proportionnelle au carré de l'amplitude de son champ électrique. Avec ce passage au carré, les phénomènes de dégradation linéaire vus dans les sections 1.4.1.1 et 1.4.1.2 (atténuation et dispersion chromatique) se transforment en des phénomènes non linéaires dont le traitement devient difficile. Contrairement à la détection directe, la détection cohérente permet d'accéder à l'amplitude et à la phase du signal optique entrant. Plusieurs avantages s'offrent étant donné que l'information concernant la phase est complètement préservée y compris la possibilité d'utiliser des formats de modulation plus élaborés. La sensibilité du récepteur est aussi améliorée pour ce type de détection. De plus, le récepteur cohérent est capable de récupérer un signal ayant n'importe quelle longueur d'onde chose qui était impossible avec les anciens récepteurs qui ne pouvaient recevoir que la longueur d'onde correspondante à leurs filtres. Le principe de fonctionnement de la détection directe en communication optique est similaire à celui en communication Radio Fréquence (RF), domaine de son apparition. Cependant, la fréquence en optique est largement supérieure à celle en RF ce qui fait qu'il y a une différence dans les composants et configurations de circuits entre les deux domaines. Un récepteur cohérent est muni d'un oscillateur local émettant un signal sur une fréquence donnée réglable. Le signal optique reçu multiplié par le signal de l'oscillateur génère un signal résultant ayant la somme et la différence des fréquences de ces deux derniers signaux. Un filtre passe-bas est généralement utilisé pour éliminer la composante somme des fréquences. Le signal en bande de base peut ainsi être récupéré en cas d'égalité de sa fréquence avec la fréquence de l'oscillateur local.

Le module de traitement du signal DSP :

L'un des plus importants avantages de la communication optique cohérente est la possibilité de compenser les phénomènes de dégradation affectant la fibre en utilisant un module de traitement de signal abrévié par DSP. L'ensemble des fonctionnalités implémentées par un module DSP peuvent être divisées en des algorithmes d'égalisation et ceux de synchronisation (Savory, 2013). Le premier type d'algorithmes vise principalement à compenser les phénomènes de dégradation du canal et les imperfections causées par l'émetteur et le récepteur. L'égalisation contient les opérations allant de la correction de synchronisation et de l'amplitude du signal sortant du convertisseur analogique-numérique qui fonctionne à haute vitesse jusqu'à la

compensation numérique de la dispersion chromatique, dispersion des modes de polarisation et la dispersion chromatique résiduelle. Une fois le signal égalisé, les algorithmes de synchronisation interviennent pour aligner à la fois les oscillateurs électriques et optiques minimisant ainsi l'impact de la différence de fréquence et phase entre l'émetteur et le récepteur.

1.6.2 Principe de fonctionnement d'un système optique cohérent DP-QPSK

L'architecture de base d'un système de communication optique cohérent modulé en quadrature de phase et multiplexé sur les deux polarisations de la lumière est illustrée dans la figure 1.5. Trois grands blocs sont schématisés : le transmetteur, la fibre optique et le récepteur cohérent. À l'émission, les signaux clients à 100 Gbits/s venant de différents canaux optiques (nommé « *Tributary* » sur la figure) sont multiplexés sur le même signal. Après, quelques informations utiles au code correcteur d'erreurs (FEC) sont ajoutées au signal numérique. Pour ce type de modulation, chaque symbole représente quatre bits de données. De ce fait, le signal sera transmis sous 4 flux de données à 28 Gbauds aux modulateurs. Le laser code les flux de bits reçus en phase, en quadrature de phase et sur les deux polarisations orthogonales X et Y. Un autre composant optique nommé PBC (*Polarization Beam Combiner* : PBC) fera la combinaison des deux polarisations pour les transmettre ensuite sur la même liaison de fibre optique. Cette dernière est formée d'une cascade de tronçons de fibres contenant des EDFA et des ROADM afin d'ajouter et/ou extraire les longueurs d'onde voulues (même architecture que celle présentée dans la figure 1.3. Du côté du récepteur, le signal optique est divisé dans les directions de polarisations X et Y par le diviseur de polarisation (*Polarization Beam Splitter* : PBS) qui joue le rôle de référence de polarisation locale. Le signal est ensuite combiné avec un laser servant d'oscillateur local. L'étape suivante sera de diviser chaque polarisation du signal et l'associer à chacune des deux phases orthogonales polarisées venant de l'oscillateur local. Dans le but de fournir quatre signaux optiques aux photodiodes, plusieurs équipements sont introduits à l'architecture y compris des diviseurs optiques, des déphaseurs (*Phase Shift*) et des coupleurs. Quant au module '*Agile Engine*', il assure la conversion analogique en numérique du signal et les fonctions de traitement de signal de signal, DSP. Finalement, comme expliqué auparavant,

le FEC permet de détecter et corriger quelques erreurs de transmission tout en se servant des informations ajoutées en émission (Roberts, O'Sullivan, Wu, Sun, Awadalla, Krause & Laperle, 2009). Les signaux des clients sont retrouvés dans le bloc '*Tributary*'.

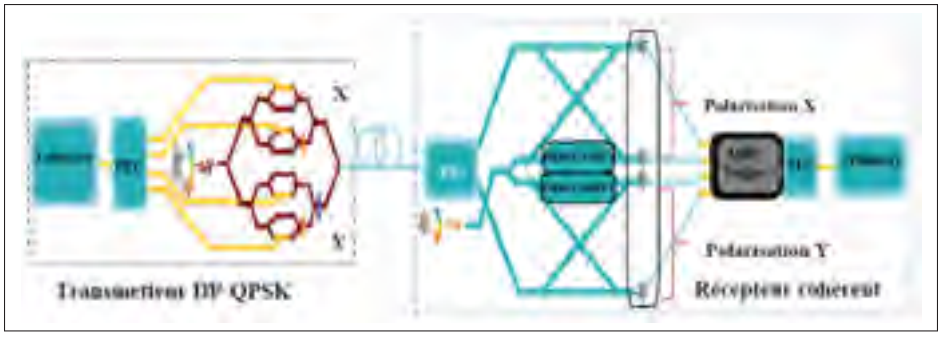


Figure 1.5 Schéma simplifié d'un système optique cohérent modulé en DP-QPSK
Tirée de Yaméogo (2017)

1.7 Le monitoring dans les réseaux optiques cohérents

La communication par fibre optique a connu une croissance éminente au cours des dernières années en résultat de l'augmentation incessante de la demande en bande passante. À son tour, cette demande est stimulée par la croissance du trafic internet que ce soit en termes du nombre d'utilisateurs actifs ou en quantité de bande passante consommée par chacun d'eux. Pour répondre à ce besoin urgent, l'architecture des réseaux optiques a considérablement évolué offrant ainsi davantage de flexibilité et une meilleure capacité de transmission. La performance d'un réseau optique opérant avec des débits élevés et sur de longues distances dépend fortement de l'étendue des dégradations introduites dans les signaux optiques transmis. Une compensation entière de certaines dégradations de nature stochastique s'avère difficile, voire impossible. Afin de garantir une bonne expérience client, les opérateurs sont obligés de conserver une marge de sécurité considérable lors de la conception des réseaux optiques entrainant ainsi le gaspillage de ressources assez précieuses. Ils peuvent aussi être amenés à utiliser des composants plus chers pour minimiser les dégradations affectant leurs réseaux. De plus, les opérations de

maintenance dans les réseaux optiques requièrent des interventions manuelles et réactives. Afin de réduire les coûts de maintenance, d'assurer une exploitation optimale des ressources et de garantir une gestion adéquate et autonome, les opérateurs ont tout intérêt à monitorer en continu les performances de leurs réseaux optiques cohérents. La surveillance ou le monitoring des performances optiques (*Optical Performance Monitoring* : OPM) correspond à l'estimation et l'acquisition de plusieurs paramètres physiques des signaux transmis dans la fibre et de divers composants constituant un réseau optique (Dong, Khan, Sui, Zhong, Lu & Lau, 2015). Ces paramètres sont généralement collectés par le système de gestion de réseau (*Network Management System* : NMS) à des intervalles réguliers (15 min dans ce projet). Ils permettent une compréhension approfondie du fonctionnement de réseau et de l'évolution des dégradations physique à tout moment.

1.7.1 Défis et perspectives du Monitoring

Le monitoring des performances des réseaux optiques vise à surmonter plusieurs défis (Chan, 2010) y compris :

- assurer une visibilité en temps réel de l'état actuel du réseau facilitant ainsi la prédiction des effets de dégradation dans un réseau à très haut débit ;
- localiser l'emplacement des dégradations afin de permettre la réparation du réseau et le reroutage du trafic internet dans les plus brefs délais. Ceci peut se faire sans avoir recours à l'installation d'équipements de monitoring coûteux à de petites distances ;
- allouer dynamiquement les ressources et permettre un changement automatique des paramètres de réseau à savoir le format de modulation, la longueur d'onde du canal, le gain de l'amplificateur, etc. ;
- diagnostiquer le réseau et empêcher les pannes de services majeures de se produire. En tout, l'OPM a pour perspective de rendre le réseau intelligent, proactif et capable d'observer et d'analyser au mieux ses données de monitoring pour ajuster ses paramètres de transmission. (Chan, 2010).

1.7.2 Paramètres monitorés dans les réseaux optiques cohérents

Les paramètres qui ont été monitorés dans le cadre de ce projet peuvent être classifiés en quatre grandes catégories :

- les puissances optiques à savoir la puissance transmise, la puissance reçue, la puissance à l'entrée et à la sortie de chaque amplificateur ;
- les dégradations physiques à savoir le délai de groupe différentiel et les pertes par réflexion optiques (*Optical Return Loss* : ORL). Dans ce dernier, il s'agit de la perte de puissance du signal résultant de la réflexion qui est à son tour due à une discontinuité dans une fibre optique ;
- les erreurs, des statistiques sur la correction d'erreurs et le déni de services à savoir le nombre de secondes erronées ou non disponibles dans la couche Ethernet ou dans une unité de donnée, le FEC, le nombre de violations de code, le nombre d'erreurs corrigées en une seconde, etc. ;
- le BER et ses variantes à savoir le pre-FEC BER, post-FEC BER et le facteur Q.

Pour les niveaux de puissances, les dégradations physiques et le BER ainsi que ses variantes précédemment nommées, il s'agit de mesurer ou estimer régulièrement la valeur maximale, minimale et moyenne sur un intervalle de temps donné.

CHAPITRE 2

PRÉDICTION DE LA QUALITÉ DE PERFORMANCE DANS LES RÉSEAUX OPTIQUES

2.1 Introduction

Ce chapitre passe en revue les connaissances de base sur les séries temporelles directement reliées avec cette recherche. Vient ensuite la présentation des modèles qui seront appliqués dans ce projet. L'apprentissage par transfert est aussi exploré. Finalement, les principaux travaux menés dans des domaines connexes sont discutés.

2.2 Les séries temporelles

Cette section définit une série temporelle et présente ses composantes. Les principaux éléments d'analyse et caractérisation des séries temporelles sont aussi expliqués.

2.2.1 Définition des séries temporelles

Une série temporelle est définie comme étant une suite d'observations d'une variable y mesurée de façon ordonnée dans le temps. Si la série contient des observations pour plusieurs variables, elle est dite multivariée. Dans le cas contraire, il s'agit d'une série temporelle univariée. Elle peut être aussi continue ou discrète. Dans une série temporelle discrète, les observations sont faites sur des moments discrets, tandis que dans une série continue elles sont enregistrées de façon continue dans un intervalle de temps donné. Ce projet se concentre uniquement sur les séries temporelles discrètes. Ces dernières peuvent être représentées mathématiquement par :

$$y = \{y(t); t \in \mathbb{N}\} \tag{2.1}$$

Où :

$y(t)$ représente la valeur de la variable y à la date d'observation t.

2.2.2 Composantes des séries temporelles

Une série temporelle contient généralement une tendance, une variation saisonnière, une variation cyclique ainsi qu'une variation irrégulière (Hyndman & Athanasopoulos, 2018). Une brève description de ces quatre composants est donnée dans ce qui suit.

Tendance :

Une tendance notée $T(t)$ existe quand il y a une augmentation, une diminution ou une stabilisation à long terme dans la variable observée. En d'autres termes, la tendance traduit le comportement 'moyen' de la série temporelle. Ce n'est toutefois pas exigé que la tendance soit linéaire. Dans notre cas, aucune tendance linéaire n'a été découverte dans les données. Il n'y a pas de techniques bien spécifiques pour identifier la tendance dans une série temporelle. Cependant, si la tendance est strictement croissante ou décroissante, il est facile de l'identifier visuellement avec une représentation temporelle. Dans le cas où les données contiennent beaucoup de valeurs aberrantes, il faut procéder au lissage de la série temporelle avant d'aborder l'identification de la tendance.

Saisonnalité :

Plusieurs facteurs saisonniers à savoir la période de l'année et/ou le jour de la semaine peuvent influencer la série temporelle et donner lieu à une variation saisonnière. La saisonnalité notée $S(t)$ apparait généralement à une fréquence fixe et connue. Une faible saisonnalité journalière a été identifiée dans quelques circuits optiques aériens. Cette dernière peut être due à l'effet jour, nuit dans les séries temporelles de SNR. Certains relient l'effet jour, nuit aux travaux de réparation qui s'effectuent pendant le jour.

Cycle :

Une variation cyclique notée $C(t)$ se produit lorsque la variable observée présente des mouvements d'augmentation et de diminution à une fréquence irrégulière. De façon générale, la durée moyenne d'un cycle est beaucoup plus longue que celle de la composante saisonnière. De plus, les mouvements durant un cycle ont tendance à avoir plus de variabilité que pendant la composante saisonnière. N'ayant qu'une seule année de donnée, la variation cyclique ne peut pas être observée dans nos données.

Irrégularité :

La composante irrégulière est appelée aussi bruit ou résidus. Elle est notée par $I(t)$. Il s'agit de quelques fluctuations irrégulières, de nature aléatoire et de faible intensité. Dans notre cas, la composante irrégulière est présente dans la quasi-majorité des circuits optiques étudiés. Le bruit peut apparaitre suite à des pannes de services et des interventions humaines.

2.2.3 Analyse et caractérisation des séries temporelles

Dans cette section, l'analyse de la dépendance dans une série temporelle est présentée en premier lieu. Ensuite, l'étude de la stationnarité à l'aide des tests statistiques est expliquée. Finalement, la mesure de similarité entre les séries temporelles est explicitée.

2.2.3.1 Dépendances temporelles

Les facteurs externes ne sont pas les seuls à influencer le comportement d'une série temporelle. Cette dernière dépend aussi de ses états passés. Généralement, la dépendance temporelle est analysée à l'aide des fonctions d'autocorrélation et d'autocorrélation partielle. Ces statistiques mesurent la relation linéaire entre la variable observée et ses valeurs décalées dans le temps. Ainsi, une fonction d'autocorrélation d'ordre k quantifie la relation linéaire de la variable observée à l'instant t et sa valeur à l'instant $(t-k)$. À des fins de modélisation et de prédiction des séries temporelles, il est intéressant de représenter ces statistiques pour différents ordres k .

Fonction d'autocorrélation (ACF) :

Cette fonction mesure l'autocorrélation entre la série temporelle et tous ses états passés d'ordres k. Ainsi, l'ACF garde l'influence des valeurs $y(t-k-i)$ pour tout $i > k$. Par définition, la fonction d'autocorrélation d'ordre k notée $r(k)$ s'écrit comme suit :

$$r(k) = \frac{\sum_{t=1}^{n-k}(y(t)-\bar{y}) \times (y(t+k)-\bar{y})}{\sum_{t=1}^{n}(y(t)-\bar{y})^2} \tag{2.2}$$

Il s'agit de diviser la covariance entre les séries temporelles $y(t)$ et $y(t-k)$ par la variance de la série $y(t)$. Dans l'équation 2.2, les termes n et $\bar{y}$ correspondent respectivement à la taille de la

série temporelle (nombre d'échantillons disponibles de la variable observée) et à l'estimation de la moyenne de $y(t)$. La représentation de $r(k)$ pour différentes valeurs de k est appelée corrélogramme.

Fonction d'autocorrélation partielle (PACF) :
Quant à elle, la fonction PACF mesure la corrélation entre $y(t)$ et son retard $y(t-k)$ tout en retirant l'influence des valeurs $y(t-k-i)$, pour tout $i > k$. La représentation du PACF pour différentes ordres k est nommée corrélogramme partiel.

2.2.3.2 Stationnarité des séries temporelles

Au sens large, un processus est dit strictement stationnaire lorsque la distribution de n'importe quel ensemble de variables formant ce processus est indépendante du temps. Cependant en pratique, l'hypothèse de la stricte stationnarité n'est pas toujours nécessaire. Plutôt, une forme plus légère de stationnarité est considérée : il s'agit d'une faible stationnarité. Mathématiquement, un processus $\{Y(t), t \geq 0\}$ possède une faible stationnarité si son espérance ainsi que ses autocovariances sont invariantes par translation dans le temps.

$$\forall t \in \mathbb{N},\ E[y(t)] = \mu \tag{2.3}$$

$$\forall t \in \mathbb{N},\ \mathrm{Cov}(Y(t), Y(t+k)) = \mathrm{Cov}(Y(0), Y(k)) \tag{2.4}$$

Où :
μ est une constante et k est un intervalle de temps. Par conséquent, la variance est constante pour un processus ayant une faible stationnarité quelques soit la date t. Finalement, si une série temporelle $y(t)$ est la réalisation d'un processus stationnaire, la série elle-même est stationnaire.

En général, la non-stationnarité rend la prédiction de la variable observée une tâche difficile étant donné que les caractéristiques statistiques de la série sont variables au cours du temps. Ainsi, il faut stationnariser les séries temporelles avant d'aborder la prédiction.

Pour vérifier la stationnarité, il suffit de faire une représentation temporelle de la variable observée. Si ses propriétés ne dépendent pas du temps d'observation, il s'agit d'une série stationnaire. D'autres méthodes statistiques plus sophistiquées sont également utilisées dans ce contexte. Les plus populaires sont les tests ADF (*Augmented Dickey Fuller*) et KPSS (*Kwiatkowski-Phillips-Schmidt-Shin*).

Test ADF :

Ce test permet de déterminer l'existence d'une racine unitaire (*unit root*) dans la série temporelle facilitant ainsi de conclure quant à la stationnarité. En fait, cette dernière composante correspond à une tendance stochastique, autrement dit un comportement aléatoire suivi d'une dérive des valeurs observées. Son existence rend la série imprévisible. L'hypothèse nulle suppose la présence d'une racine unitaire. Valider cette hypothèse revient à dire que la série est non stationnaire. Le résultat de test, le 'p-value' et les valeurs critiques à des intervalles de confiance de 1%, 5% et 10% sont obtenus en appliquant l'ADF. Si le résultat de test est inférieur à la valeur critique indiquée, l'hypothèse nulle est rejetée avec un intervalle de confiance donnée. La série est donc stationnaire.

Test KPSS :

Le KPSS est un autre test statistique qui sert à vérifier la stationnarité des séries temporelles autour d'une tendance moyenne ou linéaire. L'hypothèse nulle suppose une stationnarité faible ou la présence d'une tendance déterministe dans la série (Hadri & Rao, 2009). Cependant, l'hypothèse alternative admet la présence d'une racine unitaire d'où la non-stationnarité. Dans le cas où la statistique du test est supérieure à la valeur critique, l'hypothèse nulle est rejetée. Le KPSS identifie une série comme étant stationnaire uniquement en absence de racine unitaire.

Généralement les deux tests sont appliqués pour tester la stationnarité. Cependant, leurs résultats peuvent être contradictoires. Dans le tableau 2.1, les différents cas en pratique sont expliqués.

La différenciation implique l'application de la différence entre termes consécutifs de la série temporelle. Il est possible d'effectuer plusieurs différenciations de façon répétée (n fois) jusqu'à obtenir une série stationnaire. Quant à la transformation de la série, elle vise la stabilisation de la

Tableau 2.1 Étude de la stationnarité avec les tests ADF et KPSS

	Cas 1	**Cas 2**	**Cas 3**	**Cas 4**
Résultat de l'ADF	Non-stationnaire	Stationnaire	Non-stationnaire	Stationnaire
Résultat du KPSS	Non-stationnaire	Stationnaire	Stationnaire	Non-stationnaire
Conclusion sur les tests	Non-stationnaire	Stationnaire	Existence d'une tendance	Différence stationnaire
Comment stationnariser la série ?	Différenciation ou transformation de la série	—	Éliminer la tendance	Différencier la série

variance. Ces transformations incluent l'application du logarithme, racine carrée sur les valeurs observées, etc.

2.2.3.3 Similarité entre les séries temporelles

Deux séries temporelles peuvent avoir une certaine similarité entre elles. La recherche ou la mesure de similarité est une étape indispensable dans plusieurs applications y compris la classification des séries temporelles ou encore dans le transfert d'apprentissage pour les séries temporelles. Plusieurs méthodes servent à quantifier la similarité dans le domaine temporel. La distance entre deux séries temporelles de la même taille N, $y(t) = y(1), y(2), \ldots, y(N)$ et $z(t) = z(1), z(2), \ldots, z(N)$ correspond à la longueur du chemin reliant la paire de points. Plus la distance est grande, plus la similarité entre les deux séries est minime. La distance de *Minkowski* est l'une des plus populaires. Elle est exprimée par l'équation 2.5.

$$D_{\text{Minkowski}}\big(y(t), z(t)\big) = \Big(\sum_{t=1}^{N} |(y(t) - z(t)|^P \Big)^{\frac{1}{P}} \tag{2.5}$$

Si P est égale à un, la distance est appelée distance de *Manhattan*. Dans le cas où P est égale à deux, cette dernière est nommée distance Euclidienne. La principale limitation par rapport aux distances précédemment mentionnées est qu'elles sont appliquées pour deux séries

ayant la même longueur de séquences. Cependant, en pratique les séries temporelles peuvent avoir des tailles différentes. Pour faire face à cet inconvénient, plusieurs méthodes ont été suggérées. La plus connue est celle de déformation dynamique temporelle (*Dynamic Time Wrapping* : DTW) Elle a été proposée par (Berndt & Clifford, 1994). La particularité de cette méthode est qu'elle permet de comparer chaque point de la première série avec un ou plusieurs points décalés dans le temps de la deuxième série. Ainsi, elle permet de trouver des similarités entre les deux séries temporelles même si elles apparaissent à des instants décalés. Le principe du DTW est expliqué conformément à (Keogh & Pazzani, 2001). Supposons que $y(t) = \{y(1), y(2), \ldots, y(N)\}$ et $z(t) = \{z(1), z(2), \ldots, z(M)\}$ sont deux séries temporelles de taille N et M respectivement. Supposons aussi que D est une matrice de taille $N \times M$ où le ($i^{\text{ème}}$, $j^{\text{ème}}$) contenant la distance $d(y_i, z_j)$ entre les deux points y_i et z_j. La distance Euclidienne est typiquement utilisée. L'alignement point à point et la relation de correspondance entre les séries temporelles $y(t)$ et $z(t)$ peuvent être représentés par un chemin de déformation temporelle (*wrapping path*) noté W, schématisé dans la figure 2.1 ci-dessous. Ce dernier est exprimé par :

$$W = \langle w_1, w_2, \ldots, w_K \rangle \tag{2.6}$$

Où $\max(m, n) \leq K < M + N + 1$

Le k-ième élément $w_k = (i, j)_k$ indique l'alignement entre les points y_i et z_j. Le chemin de déformation temporelle est sujet à plusieurs contraintes expliquées dans la référence mentionnée : conditions aux limites, continuité et monotonicité. Le chemin optimal satisfait l'équation 2.7. La division par K permet de compenser le fait que les chemins peuvent avoir des tailles différentes.

$$DTW(y(t), z(t)) = \min_W \left\{ \frac{1}{K} \sqrt{\sum_{k=1}^{K} w_K} \right\} \tag{2.7}$$

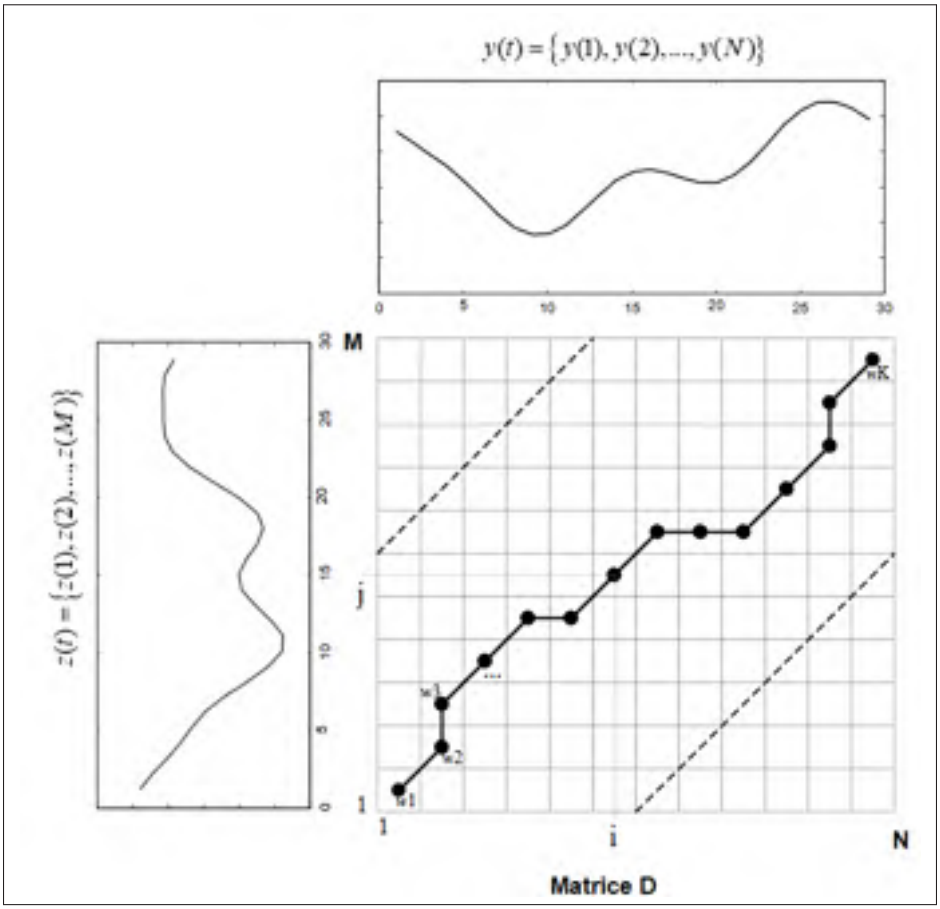


Figure 2.1 Un exemple de *wrapping path*
Adaptée de Keogh & Pazzani (2001)

2.3 Les modèles de prédiction linéaire ARIMA et SARIMA

En général, les modèles de prédiction linéaires les plus populaires pour les séries univariées sont les variantes du modèle autorégressif intégré à moyenne mobile ARIMA (*Auto Regressive lntegrated Moving Avenage*), soit les modèles de type AR, MA, ARMA, et SARIMA. Dans un modèle autorégressif (AR), la variable d'intérêt est prédite en utilisant une combinaison linéaire de ses états passés. Un modèle AR d'ordre p noté AR(p) s'écrit sous la forme :

$$y(t) = \epsilon(t) + \sum_{i=1}^{p} \phi_i y(t-i) \tag{2.8}$$

Où :

ϕ_i, tel que $1 \leq i \leq p$ sont les coefficients du modèle AR de type réel et $\epsilon(t)$ correspond à un bruit blanc de variance σ^2. Le but ici est de déterminer la valeur optimale de p qui capture le mieux l'ordre du modèle autorégressif et de trouver les coefficients du modèle.

Le modèle à moyenne mobile (MA), introduit pour la première fois par (Slutzky, 1937), prédit la variable d'intérêt en utilisant une combinaison linéaire des erreurs de prédiction passées. Chaque valeur est prédite à l'aide d'une moyenne mobile, d'où le nom du modèle. Un modèle MA d'ordre q noté MA(q) s'écrit :

$$y(t) = \epsilon(t) + \sum_{i=1}^{q} \theta_i \epsilon(t-i) \tag{2.9}$$

Où :

θ_i, tel que $1 \leq i \leq q$, sont les coefficients du modèle MA de type réel et $\epsilon(t)$ correspond à un bruit blanc de variance σ^2. Ce modèle est développé de façon itérative tout en supposant une certaine valeur q de termes d'erreurs utilisé dans la prédiction de valeurs futures. La combinaison des deux modèles AR(p) et MA(q) donne le modèle ARMA(p,q). Une série temporelle suit un processus ARMA si elle est stationnaire. Elle s'écrit sous la forme suivante pour tout instant t :

$$y(t) = \phi_1 y(t-1) + \phi_2 y(t-2) + \ldots + \phi_p y(t-p) + \epsilon(t) + \theta_1 \epsilon(t-1) + \theta_2 \epsilon(t-2) + \ldots + \theta_q \epsilon(t-q) \tag{2.10}$$

Les $p + q + 1$ paramètres de l'ARMA peuvent être estimés à partir des observations disponibles. Généralement, le corrélogramme et le corrélogramme partiel sont utilisés pour estimer p et q. La méthode d'estimation à maximum de vraisemblance (*Maximum-Likelihood Estimation* : MLE) est aussi utilisée pour déterminer les paramètres θ_i et ϕ_i . Dans le cas où la série temporelle $y(t)$ ne satisfait pas l'hypothèse de la stationnarité, une différenciation à l'ordre d est appliquée. Ce dernier paramètre d correspond au nombre de différentiation requises pour stationnariser les données. Ainsi, la série suit un processus ARIMA (p,d,q) qui est un type spécial d'ARMA où la

différentiation est prise en compte. Ce modèle s'écrit sous la forme suivante :

$$y_d(t) = \phi_1 y(t-1)+\phi_2 y(t-2)+\ldots+\phi_p y(t-p)+\epsilon(t)+\theta_1\epsilon(t-1)+\theta_2\epsilon(t-2)+\ldots+\theta_q\epsilon(t-q)+\epsilon(t) \tag{2.11}$$

Où :

$y_d(t)$ est la série différenciée à l'ordre d.

Finalement, le modèle SARIMA (*Seasonal* ARIMA) est utilisé quand la série présente une saisonnalité. Il est noté SARIMA(p,d,q)(P,D,Q,m). À part les paramètres p, d et q, plusieurs autres viennent s'ajouter à ce modèle : P, D, Q et m. Les paramètres p, d, q, P, D et Q sont respectivement les ordres des composantes non saisonnières et saisonnières de la série temporelle. Le paramètre m fait référence au nombre de périodes dans chaque saison dans les données disponibles.

2.4 Apprentissage automatique pour la prédiction temporelle

Plusieurs techniques d'apprentissage automatique peuvent être utilisés dans la prédiction des séries temporelles. Commençons par introduire la notion d'apprentissage automatique ou encore apprentissage artificiel. Ce dernier est défini par (Mitchell et al., 1997) comme suivant : un programme apprend d'une expérience E pour effectuer des tâches données avec une certaine mesure de performance P si sa performance à effectuer ces tâches, mesuré par P s'améliore à l'aide de l'expérience E. Dans cette recherche, la tâche est une prédiction temporelle supervisée : des modèles sont entraînés à prédire la qualité de performance dans des circuits optiques opérationnels pour un horizon de prédiction spécifique. La mesure de performance se fait à l'aide de plusieurs métriques y compris la racine carrée de l'erreur quadratique moyenne, le coefficient de détermination, etc. L'ensemble de ces métriques sera détaillé dans la section 3.5. Quant à l'expérience, elle est acquise avec les observations de la base de données. Lors de l'entraînement supervisé, le modèle se base sur des observations aux instants précédents dont les valeurs prédites aux instants futurs sont connues pour apprendre à prédire de nouveaux exemples non étiquetés. Il vise donc à apprendre une fonction permettant le mappage entre les entrées et

les sorties cibles. Tous les patrons appris par le modèle dérivent nécessairement de la base de données d'entraînement.

2.4.1 Les modèles

Les modèles d'apprentissage automatique évalués pour prédire la qualité de performance dans les réseaux optiques opérationnels sont présentés dans cette section.

2.4.1.1 Notions générales sur les réseaux de neurones

Les réseaux de neurones représentent l'un des modèles les plus utilisés en apprentissage automatique. Ils sont essentiellement inspirés des réseaux de neurones biologiques. Un neurone biologique reçoit en entrée des influx nerveux (autrement des signaux provenant d'autres neurones), les traite en donnant à chacun un poids relatif à son importance et génère un autre signal en sortie. De façon analogique, un neurone artificiel calcule le produit entre les poids (notés w_i) et les signaux d'entrées (notés x_i) correspondants, fait la somme de ces produits puis ajoute un biais (noté b). Le tout sera passé à travers une fonction d'activation (notée f) qui ajoute de la non-linéarité (figure 2.2). Plus de détails sur les fonctions d'activations évaluées dans ce projet sont fournies dans l'annexe I. Un neurone en soi ne peut pas réaliser des tâches aussi complexes. Plutôt, c'est l'interconnexion de plusieurs neurones qui permet de le faire. Ainsi un réseau de neurones est un modèle connexionniste qui réalise un traitement global complexe en assemblant plusieurs traitements élémentaires simples. Les réseaux de neurones sont formés de plusieurs couches successives. La première couche est appelée couche d'entrée et la dernière est celle de sortie. Quant aux couches intermédiaires, elles sont les couches cachées. En ayant plus de deux couches cachées, le réseau de neurones est profond. Il est capable d'extraire des patrons plus complexes lors de l'entraînement. Un réseau de neurones est un système adaptatif qui peut ajuster ses poids en fonction des informations qui lui sont transmises durant la phase d'entraînement, et ce de façon itérative : durant cette phase, le réseau de neurones prédit à chaque fois les sorties en fonction des entrées et des poids qu'il possède. Connaissant déjà les sorties attendues, il calcule l'erreur entre ces dernières et les sorties calculées. Il optimise ensuite ses

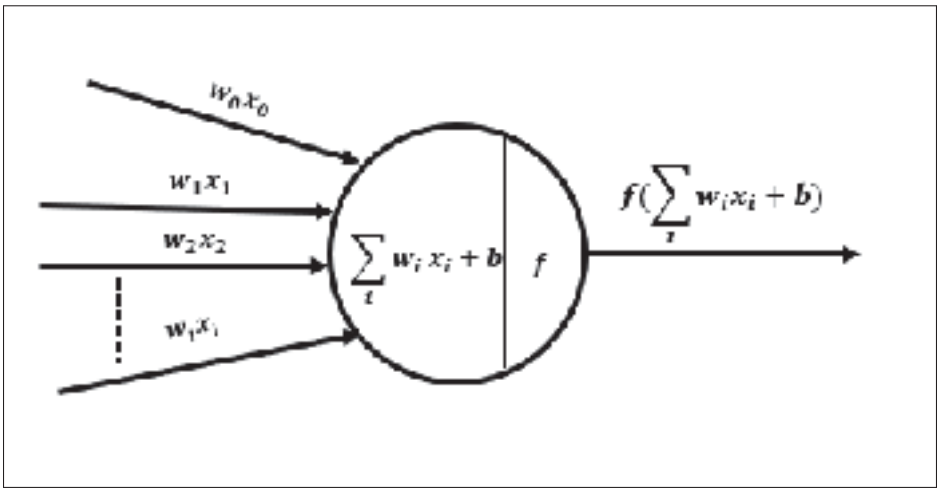


Figure 2.2 Fonctionnement d'un neurone artificiel

poids afin de minimiser l'erreur de prédiction et améliorer par conséquent les résultats suivants. La rétropropagation est un exemple d'algorithme qui permet de minimiser cette erreur. Elle sera détaillée dans la section 2.4.2.

Depuis l'apparition des neurones formels dans les années 1940, plusieurs types de réseaux de neurones avec des propriétés aussi diverses ont été développés. Deux grandes catégories peuvent être distinguées : les réseaux de neurones récurrents et les réseaux de neurones à propagation avant. Les réseaux de neurones convolutifs font partie de ceux à propagation avant. Ces derniers seront détaillés dans les deux sections suivantes.

2.4.1.2 Les réseaux de neurones récurrents

Les réseaux de neurones récurrents (*Recurrent Neural Network* : RNN) ont été historiquement dédiés à la modélisation des séquences (Goodfellow, Bengio & Courville, 2016), autrement dit des signaux de taille variable. Les premières recherches ont tenté d'utiliser les RNNs dans le traitement de langage naturel (Young, Hazarika, Poria & Cambria, 2018). Des résultats promoteurs ont été obtenus. L'application des RNNs dans la prédiction des séries temporelles a aussi suscité de l'intérêt, et ce dans plusieurs domaines d'application (Rangapuram, Seeger, Gasthaus, Stella, Wang & Januschowski, 2018 ; Salinas, Flunkert, Gasthaus & Januschowski,

2019). Notons que les données temporelles ne sont autres que des données séquentielles. Des résultats intéressants ont également été relevés. La capacité des RNNs à modéliser la dépendance à long et à moyen terme dans les données séquentielles est derrière sa popularité. En effet, ils sont capables de prendre en considération un contexte dans leurs fonctions de décisions. Ce dernier contexte provient d'une connexion en boucle : à l'étape courante, une ou plusieurs informations des étapes précédentes sont prises en compte. En d'autres termes, l'état de mémoire est mis à jour de façon récursive en utilisant de nouvelles observations à chaque instant t comme c'est indiqué dans la figure 2.3. Mathématiquement, un réseau récurrent possède une variable d'état

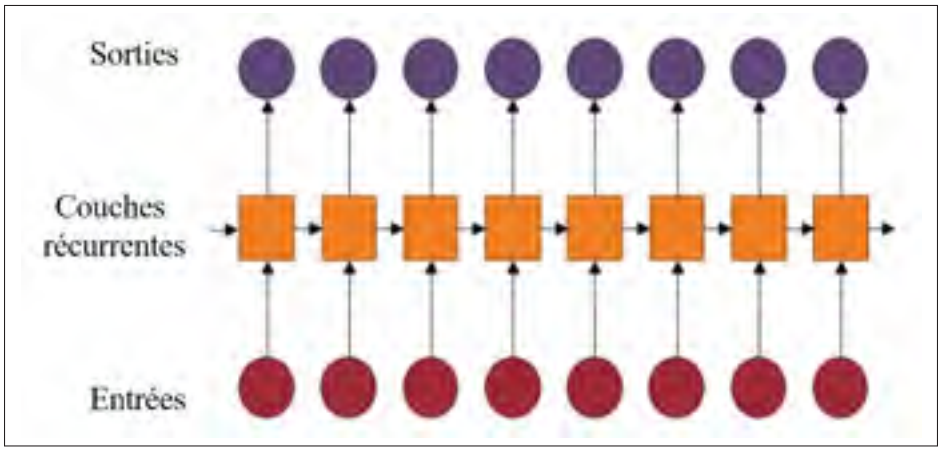


Figure 2.3 Modèle de RNN simplifié
Adaptée de Lim & Zohren (2020)

notée $h(t)$ exprimée comme suit :

$$h(t) = f(h(t-1), x(t); w_f) \tag{2.12}$$

Où :

$x(t)$ représente l'entrée du réseau de neurones observée à l'instant t. La sortie $s(t)$ est calculée en fonction de la variable d'état $h(t)$:

$$s(t) = g(h(t); w_g) \tag{2.13}$$

Les fonctions f et g représentent une transformation affine auquel est ajoutée de la non-linéarité. Quant aux poids w_f et w_g, ils seront optimisés lors de la phase d'entraînement de façon à fournir de bonnes sorties.

Au cours des dernières décennies, de nombreuses architectures de RNN ont été développées. Certains sont simples (*vanilla* RNN) tels que les réseaux d'*Elman* (Elman, 1990) et ceux de *Jordan* (Jordan, 1997). Ces derniers utilisent la rétropropagation à travers le temps (*BackPropagation Through Time* : BPTT) pour apprendre. Ils sont ainsi sujets à la disparition et à l'explosion du gradient. Les RNNs simples ne sont pas capables de maintenir un état au cours du temps. De ce fait, ils sont restreints à la modélisation des dépendances à court ou à moyen terme uniquement (Hochreiter, Bengio, Frasconi, Schmidhuber et al., 2001). D'autres architectures beaucoup plus sophistiquées ont été développées pour adresser les limitations des RNNs simples. Les plus populaires sont les réseaux récurrents avec mémoires à court et à long termes et les réseaux GRU.

1. **Les réseaux LSTM :**

   Les réseaux avec mémoires à court et long termes plus connus sous le nom de réseaux LSTM (*Long Short Term Memory networks*) sont un type de RNNs capable d'apprendre les dépendances à long terme. Le réseau LSTM est initialement introduit par Hochreiter & Schmidhuber (1997). L'architecture a ensuite considérablement évolué (Bayer, Wierstra, Togelius & Schmidhuber, 2009 ; Gers, Schmidhuber & Cummins, 1999 ; Graves, Fernández & Schmidhuber, 2005). C'est grâce à l'existence d'une mémoire interne nommée cellule (*cell*) et notée $c(t)$ que le LSTM est apte à maintenir un état sur une longue période de temps. L'état de cette dernière cellule est régulé par trois portes de contrôle (*gates*) : porte d'entrée (*input gate*), porte d'oubli (*forget gate*) et porte de sortie (*output gate*). Elles sont respectivement notées $i(t)$, $f(t)$ et $o(t)$ comme est indiqué dans la figure 2.4. Les poids des gates peuvent être ajustés de façon à garder l'information tant qu'elle est utile uniquement. Ceci permet d'éviter le problème du gradient disparaissant. Le processus de fonctionnement d'un neurone LSTM se fait en plusieurs étapes. La première consiste à identifier l'information qui sera éliminée de la cellule mémoire précédemment initialisée à

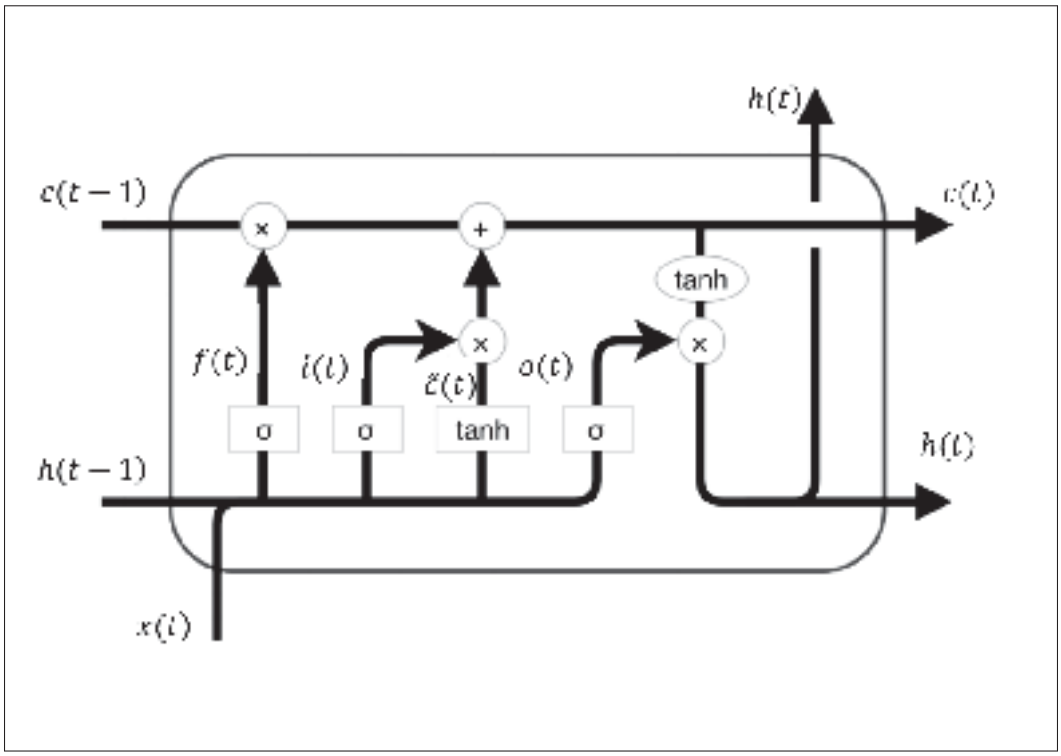


Figure 2.4 Schéma d'un neurone LSTM avec une seule cellule mémoire

$c(t-1)$. C'est la porte d'oubli qui permet entre autres de régulariser la force de la rétroaction. La décision est faite en considérant l'entrée $x(t)$, la variable d'état cachée à l'instant $t-1$, $h(t-1)$ et ce en utilisant une fonction sigmoïde σ. Cette dernière produit des valeurs entre 0 et 1. En tout, la porte d'oubli est exprimée par l'équation 2.14.

$$f(t) = \sigma\Big(\sum_j U_j^f x_j(t) + \sum_j W_j^f h_j(t-1) + b^f\Big) \tag{2.14}$$

Où :

U^f, W^f et b^f représentent respectivement les poids d'entrées, les poids récurrents et le biais de la porte d'oubli.

La deuxième étape consite à mettre à jour le contenu de la cellule mémoire à l'instant actuelle t. Deux opérations devraient se faire avant d'y procéder. D'abord, la porte d'entrée est déterminée. Cette dernière permet d'ajuster la proportion d'accumulation de caractéristiques.

Elle est donnée par l'équation suivante :

$$i(t) = \sigma\Big(\sum_j U^i_j x_j(t) + \sum_j W^i_j h_j(t-1) + b^i\Big) \tag{2.15}$$

Où :

U^i, W^i et b^i représentent respectivement les poids d'entrées, les poids récurrents et le biais de la porte d'entrée.

Ensuite, un vecteur $\tilde{c}(t)$ contenant les valeurs candidates à être ajoutées dans la cellule mémoire à l'instant t est créé en utilisant la formulation suivante :

$$\tilde{c}(t) = \tanh\Big(\sum_j U^{\tilde{c}}_j x_j(t) + \sum_j W^{\tilde{c}}_j h_j(t-1) + b^{\tilde{c}}\Big) \tag{2.16}$$

Où :

$U^{\tilde{c}}$, $W^{\tilde{c}}$ et $b^{\tilde{c}}$ représentent respectivement les poids d'entrées, les poids récurrents et le biais du vecteur candidat. La tangente hyperbolique et la multiplication élément par élément sont respectivement notées $tanh$ et '$*$' dans l'équation.

Le contenu de la cellule mémoire est mis à jour selon l'équation 2.17

$$c(t) = f(t) * c(t-1) + i(t) * \tilde{c}(t) \tag{2.17}$$

Il reste uniquement à définir la porte de sortie qui contrôle la proportion de la cellule mémoire qui sera acheminée à la sortie (équation 2.18 :

$$o(t) = \sigma\Big(\sum_j U^o_j x_j(t) + \sum_j W^o_j h_j(t-1) + b^o\Big) \tag{2.18}$$

Où :

U^o, W^o et b^o représentent respectivement les poids d'entrées, les poids récurrents et le biais de la porte de sortie.

Finalement, la variable d'état à l'instant actuel t, $h(t)$ est donnée par l'équation 2.19 :

$$h(t) \;=\; o(t)\tanh\big(c(t)\big) \tag{2.19}$$

2. **Les réseaux GRU :**

Le réseau de neurones GRU (*Gated Recurrent Unit*) est une version simplifiée du LSTM. Elle a été récemment introduite par (Cho, Van Merriënboer, Bahdanau & Bengio, 2014). Les deux derniers réseaux ont des performances équivalentes (Chung, Gulcehre, Cho & Bengio, 2014). Cependant, le GRU est plus efficace en termes de rapidité de calcul, car il possède moins de portes de contrôle comparé au réseau LSTM. Il a uniquement deux portes appelées porte de pertinence (*relevance gate*) et porte de modification (*update gate*), notées respectivement $r(t)$ et $u(t)$. Dans le contexte de GRU, il n'y a plus la séparation entre la variable d'état cachée et la cellule mémoire. Un vecteur $\tilde{h}(t)$ candidat servant à remplacer le contenu de la variable d'état à l'instant actuel t est donné par la relation suivante :

$$\tilde{h}(t) \;=\; \tanh\Big(r(t) * \big(\sum_j U_j^{\tilde{h}} x_j(t) + \sum_j W_j^{\tilde{h}} h_j(t-1) + b^{\tilde{h}}\big)\Big) \tag{2.20}$$

$r(t)$ permet de quantifier la pertinence de la variable d'état à l'instant précédent $t-1$ dans le calcul du vecteur candidat $\tilde{h}(t)$. Elle est formulée comme suit :

$$r(t) \;=\; \sigma\Big(\sum_j U_j^r x_j(t) + \sum_j W_j^r h_j(t-1) + b^r\Big) \tag{2.21}$$

Quant à la porte de modification, elle permet de déterminer la proportion de la mise à jour de $h(t)$ par $\tilde{h}(t)$. Elle est exprimée comme suit.

$$u(t) \;=\; \sigma\Big(\sum_j U_j^u x_j(t) + \sum_j W_j^u h_j(t-1) + b^u\Big) \tag{2.22}$$

La mise à jour de la variable d'état à l'instant t est donnée par l'équation suivante :

$$h(t) \;=\; u(t) * \tilde{h}(t) \;+\; \big(1-u(t)\big) * h(t-1) \tag{2.23}$$

2.4.1.3 Les réseaux de neurones convolutifs

Le réseau de neurones convolutif ou à convolution (appelés aussi CNN pour *Convolutional Neural Networks*) (LeCun et al., 1989) est un type de réseaux de neurones adaptés au traitement de données ayant une topologie de type grille. C'est le cas pour les séries temporelles (grille 1-D) ou encore les images (grille 2-D de pixels) (Goodfellow *et al.*, 2016). Le CNN a fait preuve de succès dans divers problèmes y compris la reconnaissance d'images, la classification et récemment la prédiction de séries temporelles. Intuitivement, appliquer un réseau convolutif pour la prédiction de série temporelle consiste à apprendre des filtres qui représentent les patrons récurrents dans les données et les utiliser pour prédire les valeurs futures. Le réseau convolutif utilise l'opération mathématique de convolution, d'où son nom. Dans ce qui suit, le fonctionnement de la convolution est expliqué, les couches communes formant ce réseau de neurones sont décrites et les facteurs motivant son utilisation sont détaillés.

La convolution est une opération qui consiste à faire glisser une matrice de poids appelée filtre ou aussi *Kernel* sur un signal d'entrée. À chaque position, les valeurs du filtre sont multipliées avec les valeurs d'entrées. Le tout sera ensuite additionné en une seule valeur (figure 2.5). Le filtre peut se déplacer sur le signal d'entrée d'un pas (dite aussi décalage) supérieur ou égal à un. Si le pas est égal à un, le filtre avance sur le signal d'entrée une observation à la fois (comme c'est le cas pour la figure 2.5). Cependant, s'il est plus d'un, la sortie produite est sous-échantillonnée. En général, la taille du filtre est largement plus petite que celle du signal d'entrée. La partie du signal auquel le filtre est appliqué s'appelle champ réceptif. Le résultat de la convolution du filtre sur tout le signal d'entrée est appelé carte de caractéristique ou encore *feature map*.

En tout, la convolution peut être considérée comme une mesure de corrélation entre deux tenseurs pour différents décalages possibles. La convolution permet donc d'extraire les informations pertinentes du signal d'entrée. Trois propriétés dans l'opération de convolution ont fait de ce type de réseau une option intéressante pour les données ayant une topologie de type grille (Goodfellow *et al.*, 2016).

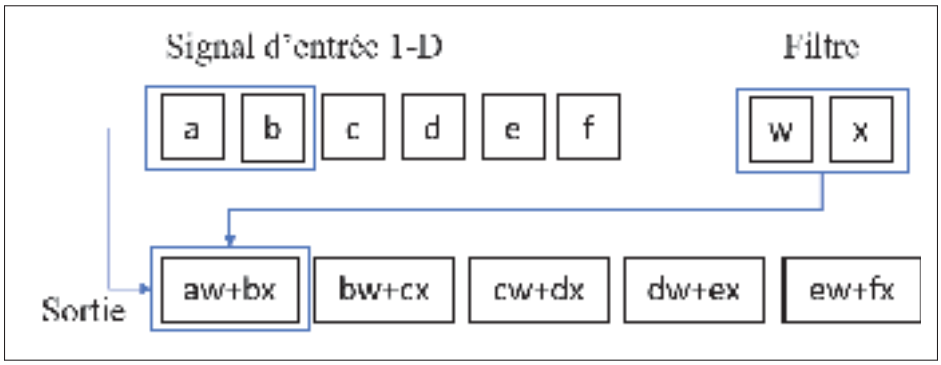


Figure 2.5 Exemple de convolution 1-D
Adaptée de Goodfellow *et al.* (2016)

1. Les interactions dans un réseau à convolution sont clairsemées. Contrairement aux réseaux de neurones traditionnels (de type perceptron multicouche) où chaque sortie est entièrement connectée aux entrées précédentes, un neurone dans un réseau convolutionnel dépend uniquement de quelques neurones dans la couche précédente. Cette propriété dérive de la taille du filtre qui est largement plus petite que le signal d'entrée. Les interactions clairsemées permettent donc de réduire les ressources en matières de mémoire, de capacité et de temps de calcul requis pour entraîner le réseau.
2. Un réseau convolutif utilise le partage de paramètres. En effet, les paramètres des filtres sont chaque fois réutilisés sur le signal d'entrée avec décalage. Cette propriété favorise ainsi l'apprentissage d'un seul ensemble de paramètres convenables pour toutes les positions au lieu d'apprendre des paramètres propres à chacune des positions. Ce partage de paramètres réduit la mémoire requise durant la phase d'entraînement.
3. La convolution est équivariante à la translation. Cette propriété est le résultat du partage des paramètres. En appliquant la convolution sur des séries temporelles, elle produit une sorte de chronologie montrant quand différents patrons apparaissent dans le signal d'entrée. En effet, si un événement est déplacé plus tard dans le temps dans le signal d'entrée, la même représentation de ce dernier apparaitra dans la sortie juste plus tard. De ce fait, le réseau convolutif est capable d'apprendre des patrons locaux invariants aux translations.

Un réseau convolutif typique est formé de deux types de couches :

Couche de convolution :

Une couche de convolution est composée de trois fonctions. D'abord, de nombreux filtres sont convolués avec le signal d'entrée en parallèle, produisant ainsi les champs réceptifs. Ensuite ces derniers sont passés à travers des fonctions d'activation pour ajouter de la non-linéarité. Les sorties de cette étape sont nommées activations. Finalement, la fonction de *pooling* est appliquée sur les activations. Dans les implémentations des réseaux convolutifs, la fonction de *pooling* est considérée comme un autre type de couche. Cette opération produit des statistiques sommaires locales sur des zones également espacées des activations reçues. *Average pooling* est un exemple de fonction de *pooling* qui retourne les moyennes locales des activations. La fonction de *pooling* réduit donc la taille des activations tout en préservant leurs caractéristiques importantes et les rendent plus invariantes aux petites translations dans le signal d'entrée Goodfellow *et al.* (2016).

Couche entièrement connectée :

Ce type de couche est connu sous le nom de *Fully Connected* (FC). Elle est généralement appliquée sur les sorties de la fonction de *pooling* préalablement aplaties. Chaque neurone est entièrement connecté aux sorties de la couche précédente. Elle apparaît comme dernière couche d'un réseau convolutif et permet de réaliser la tâche de prédiction.

Une succession de couches de convolution suivies d'une ou plusieurs couches FC peut être utilisée. L'architecture adoptée est optimisée en fonction du signal d'entrée et de la tâche réalisée. Ceci dit, une architecture optimale doit être déterminée pour chaque circuit étudié.

2.4.2 Procédure d'entraînement

Dans notre cas, la prédiction de la qualité de performance dans les réseaux optiques est modélisée comme un problème d'apprentissage automatique supervisé. Les paramètres des modèles sont toutefois optimisés en se basant sur des observations de terrain étiquetées, de façon à fournir de bonnes estimations de la variable prédite.

Pour le faire, des variantes de l'algorithme de descente du gradient sont utilisées. La descente du gradient est un processus itératif qui vise à minimiser la fonction de coût, notée $J(W)$, en

ajustant les paramètres du modèle : à chaque instant t, le gradient de la fonction de coût est calculé par rapport au vecteur de poids $W(t)$. L'inverse du gradient révèle la direction qui permet de diminuer la fonction de coût. Ensuite, le processus avance dans cette direction jusqu'à convergence vers un minimum. Il n'y aucun garantie que ce minimum soit le minimum global. Mathématiquement, le calcul du gradient se fait comme suit :

$$\Delta = -\nabla_W J(W) \tag{2.24}$$

Où :

∇ réprésente la dérivée de la fonction de coût par rapport au vecteur W.

La vitesse de convergence est contrôlée avec un hyperparamètre appelé taux d'apprentissage, noté α. Le choix de α est critique. En effet, si α est très petit, la convergence du modèle vers un optimum prendra un temps considérable. Dans le cas contraire, le modèle subira des fluctuations autour de l'optimum sans l'atteindre nécessairement. La mise à jour des vecteurs de poids à l'instant $t + 1$ est traduite par l'équation 2.25 :

$$W(t+1) = W(t) + \alpha\Delta \tag{2.25}$$

L'algorithme de descente du gradient tel qu'il est proposé par Cauchy en 1847 prend en mémoire tout l'ensemble d'entraînement à chaque itération puis applique le gradient en un seul calcul. De ce fait, l'entraînement est qualifié de lent, voire parfois impossible quand il s'agit de traiter une grande quantité de données. Cette version de l'algorithme est très contraignante en pratique. L'algorithme de descendant du gradient stochastique (Robbins & Monro, 1951) est proposé pour remédier aux limitations en mémoire : il ajuste les poids du modèle en prenant aléatoirement un exemple d'entraînement à la fois. De ce fait, chaque opération est très peu coûteuse. Cependant, la valeur calculée est une estimation peu fiable du gradient de tout l'ensemble d'entraînement. Un bon compromis entre les deux algorithmes précedemment mentionnés est de calculer le gradient sur n exemples d'entraînement sélectionnés aléatoirement à chaque itération. C'est justement ce que propose l'algorithme de descente du gradient par *mini-batch*. Le nombre d'exemples

dans chaque *mini-batch* est un autre hyperparamètre qui s'ajoute. Il est strictement inférieur aux nombre d'observations dans l'ensemble d'entraînement. Le choix du taux d'apprentissage dans cette version du gradient est difficile. En effet, une seule valeur ne convient pas nécessairement à toutes les étapes d'ajustement des poids. La fonction de coût fera des oscillations sans avoir une convergence efficace.

L'introduction du *moment* dans les algorithmes dérivés de la descente du gradient est une autre évolution importante. En effet, ce terme permet de garder en historique les directions précédemment prises par le gradient. L'idée consiste à calculer en premier lieu une moyenne mobile exponentielle des gradients précédents. Ensuite, c'est cette moyenne mobile qui est utilisée dans la formule d'ajustement des poids. Le moment permet d'améliorer la qualité de l'optimisation. En effet, la dérivé du gradient est estimée sur un *mini-batch* de données, sa direction n'est pas toujours optimale. L'ajout du moment permet donc une meilleure estimation de cette dernière. Elle permet aussi d'accélérer la phase d'apprentissage surtout si les gradients sont bruités ou lorsque la fonction de coût présente une courbure prononcée.

D'autres variantes de descente du gradient utilisent un taux d'apprentissage adaptatif ou même le combinent avec le *moment*. Adapter le taux d'apprentissage en fonction de la valeur du gradient est très avantageux. L'apprentissage devrait avancer rapidement lors des premières itérations et ralentir considérablement en s'approchant de l'optimum. Les algorithmes Adam et RMSProp seront évalués dans le cadre de ce projet. Ils sont décrits avec beaucoup plus de détails dans la section 3.7.4.

La dernière partie dans cette section se concentre sur la rétropropagation (Rumelhart, Hinton & Williams, 1986). C'est un algorithme qui se base sur la règle de la chaîne pour calculer de façon efficace les gradients d'une composition de fonctions paramétrisées. Ainsi, cet algorithme convient parfaitement aux réseaux de neurones couramment décrits comme une composition de fonctions différentiables paramétrisées. L'algorithme de rétropropagation permet donc d'utiliser la descente de gradient avec les réseaux de neurones ayant des multicouches. La rétropropagation applique une série de dérivation en chaîne pour faire propager les erreurs partant de la couche

de sortie jusqu'à la première couche cachée. Faire converger le modèle prédictif vers une configuration optimale des poids est le but ultime de la rétropropagation.

2.4.3 Apprentissage par transfert pour les séries temporelles

L'apprentissage par transfert (*transfer learning*) est une forme particulière de l'apprentissage automatique qui vise à extraire des connaissances d'une ou plusieurs tâches sources et de les appliquer à une tâche cible. D'abord, un modèle est entraîné à réaliser une certaine tâche. Ensuite, les poids de ce dernier sont réutilisés comme un point de départ dans l'entraînement d'un deuxième modèle chargé d'effectuer une autre tâche connexe.

L'apprentissage par transfert est souvent adopté avec les modèles d'apprentissage profond ayant une quantité de données insuffisante pour l'entraînement (Yosinski, Clune, Bengio & Lipson, 2014). Cette approche s'est révélée efficace dans de nombreux domaines y compris la reconnaissance d'images (Amaral, Silva, Alexandre, Kandaswamy, de Sá & Santos, 2014) et le traitement de langage naturel (Vu, Imseng, Povey, Motlicek, Schultz & Bourlard, 2014).

Récemment, elle a fait succès dans la prédiction de série financière (He, Pang & Si, 2019). Les auteurs ont proposé une nouvelle stratégie d'entraînement avec deux sources de données. Le choix de ces séries temporelles sources est crucial. Dans le cas où ces séries n'ont pas une grande similarité à la série cible de la prédiction, les performances des modèles dégradent significativement (Rosenstein, Marx, Kaelbling & Dietterich, 2005). De ce fait, il est important de bien sélectionner les séries temporelles sources. Dans la même recherche, les auteurs ont aussi proposé d'utiliser la méthode de DTW (expliquée dans la section 2.2.3.3) pour mesurer la similarité entre les séries temporelles.

L'efficacité de l'apprentissage par transfert dans la prédiction de la qualité de performance dans les réseaux optiques opérationnels avec plus qu'une source de données sera évaluée dans le cadre de ce projet. En effet, seulement douze mois de données sont disponibles pour chaque circuit étudié. Ceci se dit, un modèle utilisé pour prédire un circuit X sera préentrainé avec deux autres circuits Y et Z parmi les plus similaires à X. En d'autres termes, au lieu de commencer le

processus d'apprentissage à partir de zéro, les modèles appris lors de la prédiction des circuits Y et Z seront exploités.

2.4.4 Avancées de l'application de l'apprentissage automatique pour la prédiction de la qualité de performance

En réseaux optiques, l'apprentissage automatique a été particulièrement utilisé dans des applications connexes à la prédiction de la qualité de performance à savoir : l'identification, la localisation et la prédiction de pannes, l'estimation de la qualité de transmission (*Quality of Transmission* : QoT), le monitoring des performances, etc. Les travaux nommés dans ce qui suit ne constituent pas une liste exhaustive de ceux qui existent dans le domaine.

Dans les travaux de (Ruiz, Fresi, Vela, Meloni, Sambo, Cugini, Poti, Velasco & Castoldi, 2016), les auteurs ont proposé une méthode pour caractériser expérimentalement les défaillances de type filtrage serré (*tight filtering*) et interférence entre canaux optiques (*inter-channel interference*). Ils ont d'abord analysé les patrons des données de monitoring Pre-FEC BER et la puissance optique au niveau du récepteur et ce en cas d'apparition et d'absence des défaillances précédemment mentionnées. Ensuite, ils ont appliqué un algorithme pour localiser ces défaillances. L'algorithme de localisation s'appuie sur une approche de réseaux bayésiens. Ce dernier est entraîné pour identifier les causes les plus probables de la défaillance. Dans l'ensemble les résultats de l'algorithme sont très satisfaisants : une erreur de 0.08% a été observée pour des cas de défaillances de type filtrage serré qui ont été identifiés comme des cas normaux. Cependant, l'article s'est limité à l'analyse de deux causes de défaillances qui ne sont pas nécessairement les plus fréquentes en pratique. Dans la continuité de ces travaux, (Vela, Ruiz, Fresi, Sambo, Cugini, Meloni, Potì, Velasco & Castoldi, 2017) ont anticipé la détection de dégradation du BER dans la couche optique. Ils ont aussi visé l'identification de la défaillance en vue de faciliter sa localisation. Les auteurs ont procédé en deux étapes. D'abord, ils ont développé un algorithme nommé BANDO qui détecte les anomalies dans les données de BER (autrement les changements significatifs du BER dans les connexions optiques). Ensuite, ils ont mis en place un autre algorithme nommé LUCIDA qui permet d'identifier la cause la plus probable

de la dégradation, en cas d'apparition. Ce dernier se base sur une approche probabiliste. Pour l'évaluation des performances de ces algorithmes, différents scénarios de dégradation ont été simulés avec des équipements commerciaux. Les résultats ont montré qu'il était possible de prédire une dégradation du BER plusieurs jours à l'avance. Dans les deux derniers travaux, les résultats sont basés sur des données synthétiques et non pas de données réelles qui reflètent ce qui se passe en terrain.

La prédiction de panne est une autre application intéressante qui a bénéficié de l'apprentissage automatique. Les travaux de (Wang, Zhang, Wang, Song, Liu, Li, Lou & Liu, 2017) ont développé un algorithme basé sur le SVM (*Support Vector Machine*) et le DES (*Double Exponentional Smoothing*) permettant de prédire la défaillance d'un équipement au sein d'un réseau optique. En premier lieu, le SVM a été entraîné pour classifier un équipement optique comme *normal* ou *problématique* à partir d'un ensemble d'indicateurs physiques traduisant l'état des équipements (à savoir la puissance optique transmise et reçue, la température, etc.). En deuxième lieu, le DES est utilisé pour prédire les valeurs des indicateurs physiques après un intervalle donné. Ces valeurs servent finalement à prédire l'état de l'équipement à l'avance. Les résultats expérimentaux ont révélé une précision allant jusqu'à 95% dans la prédiction des pannes des équipements. Cette recherche a utilisé des données de terrain issues d'un opérateur pour une période de 44 jours. En tout, $14,080$ échantillons de données ont été utilisés. Cette période d'observation est aussi courte pour pouvoir généraliser les résultats. L'article s'est aussi contenté d'analyser des défaillances étroitement reliées aux états des équipements pouvant être quantifiées par les paramètres physiques monitorés.

Les travaux de (Aladin & Tremblay, 2018) ont proposé un modèle d'estimation de la qualité de transmission en se basant sur les paramètres du lien et du signal. Des effets non linéaires sont pris en compte à l'aide de formules analytiques qui se basent sur le modèle de bruit Gaussien. Dans ce projet, trois algorithmes d'apprentissage automatique ont été évalués (SVM, KNN et *Random Forest*). Les auteurs ont conclu que ces derniers sont prometteurs dans les réseaux optiques hétérogènes.

Les recherches de (Tanimura, Hoshida, Rasmussen, Suzuki & Morikawa, 2016) et (Tanimura, Hoshida, Kato, Watanabe & Morikawa, 2018) ont traité l'aspect monitoring de performance. En effet, les auteurs ont prouvé que les réseaux de neurones profonds et les réseaux convolutionnels sont capables d'estimer l'OSNR à partir d'autres paramètres physiques générés en laboratoires (*horizontal in-phase* (HI), *horizontal quadrature* (HQ), *vertical in-phase* (VI) and *vertical quadrature* (VQ)).

Un autre article (Allogba & Tremblay, 2018) a présenté une classification binaire des données de BER collectées au sein d'un réseau opérationnel. L'algorithme utilise trois caractéristiques (*features*) : la température ambiante, la période de la journée et les valeurs minimale et maximale du BER au cours de l'heure précédente. Les auteurs ont choisi le KNN (*K-Nearest Neighbors*) comme algorithme de classification. Le modèle a atteint une précision de 97.8%. Il sera intéressant de comparer les performances du KNN avec d'autres modèles de classification (à savoir le SVM et le *Random Forest*). Néanmoins, ajouter une méthode d'équilibrage de données demeure un autre aspect à approfondir.

Des travaux récents ont exploré la prédiction de la qualité de performance dans les réseaux optiques opérationnels en utilisant les techniques d'apprentissage automatique et des données de terrain et obtenu des résultats prometteurs (Aladin, Allogba, Tran & Tremblay, 2020a ; Aladin, Tran, Allogba & Tremblay, 2020b ; Tremblay, Allogba & Aladin, 2019). Des résultats promoteurs sont révélés.

CHAPITRE 3

MÉTHODOLOGIE

3.1 Introduction

Ce chapitre a pour but de présenter la méthodologie utilisée pour prédire la qualité de performance dans les réseaux optiques opérationnels en utilisant les techniques d'apprentissage automatique. D'abord, une analyse descriptive et exploratoire des circuits étudiés est fournie. Le traitement et la préparation des données sont expliqués en deuxième lieu. Selon l'étude effectuée, la puissance maximale du signal optique reçue peut être prise en compte dans la prédiction. Le chapitre décrit également la méthode de sélection des attributs explicatifs. Les critères de performances retenus comme les plus pertinents dans le cadre de cette étude sont ensuite développés. Aussi, les modèles de prédiction étant évalués sont détaillés. De plus, les différentes pratiques pour découper l'ensemble des données, entraîner les modèles, déterminer les hyperparamètres convenables, optimiser les résultats de prédiction et la stratégie d'application de l'apprentissage par transfert y sont présentées. Finalement, les outils utilisés dans le cadre de ce projet sont mentionnés.

3.2 Analyse descriptive des circuits sélectionnés

La base de données mise à la disposition de l'équipe est constituée de 12 routes de production situées dans des emplacements géographiques éloignés. Elles sont ainsi soumises à des conditions climatiques distinctes. Chaque route retrouve un ensemble de circuits transportant du trafic réel. Il y a des circuits qui relient les villes source et destination de bout en bout, d'autres relient des villes adjacentes ou proches au sein de la même route. L'étendue des circuits est diverse : longue portée (plus que 1000 km), régionale (entre 300 et 1000 km), métropolitaine (moins de 300 km) ou très courte distance (quelques km). Le déploiement de ces circuits peut se faire proche des voies ferrées ou des métros, à côté des autoroutes, en mode aérien ou enfoui dans le sol, etc. Le type de la fibre optique monomode utilisée peut également différer d'une route à l'autre et d'un circuit à l'autre : *True Wave*, *standard* NDSF (*Non Dispersion Shifted Fiber*),

etc. Pour l'ensemble des douze routes, il y a 150 circuits correspondants dont plusieurs sont bidirectionnels. Les données de monitoring sont collectées au niveau du NMS toutes les 15 minutes en utilisant des transpondeurs cohérents modulés en DP-QPSK et dont le débit est de 100 Gbit/s. La période d'observation débute en décembre 2016 et couvre une période allant jusqu'à fin novembre 2017, soit une année. Cependant, quelques circuits ont des données manquantes durant cette période d'observation ou ne sont monitorés que pour une dizaine de jours. Pour chaque mois, les données sont transmises sous la forme de fichiers *Comma Separated Values* (CSV). L'équipe a ensuite mis en place une base de données relationnelle et un tableau de bord pour faciliter l'accès, la disponibilité et l'exploration des données disponibles.

Vu le nombre élevé de circuits et la présence de données manquantes pour certains d'entre eux, des critères ont été établis pour sélectionner quatre circuits dont les résultats de test seront présentés dans ce rapport. Ceci n'empêche pas que d'autres puisse être analysés et présentés dans le cadre de ce projet. Les critères de sélection sont les suivants :

- être un circuit de bout en bout : le circuit doit parcourir les villes source et destination de la route. Un tel circuit est représentatif pour la quasi-majorité des autres circuits au sein de la même route ;
- les circuits à analyser doivent avoir des données disponibles sur toute la période d'observation précédemment mentionnée ;
- les circuits ne doivent pas avoir plus que 10% de données manquantes sur l'ensemble de la période de monitoring ;
- un des circuits à étudier doit contenir des erreurs entrainant des interruptions de services ou des pertes de données transmises ;
- les circuits à étudier doivent avoir des installations différentes : circuits aériens et enfouis dans le sol ;
- les circuits d'intérêts doivent avoir un comportement différent : circuit calme et circuit agité. Un circuit est dit calme si sa qualité de performance ne varie pas énormément durant la période d'observation et n'a pas beaucoup d'interruptions de services.

Pour satisfaire à ces critères, une étude a été faite sur l'ensemble des circuits. Elle comprend entre autres des représentations temporelles, des boîtes à moustaches de la qualité de performance, etc. Elle contient aussi des analyses et des statistiques descriptives sur les circuits. Les résultats pour les circuits d'intérêts seront présentés dans ce qui suit. Ces circuits sont désignés par : circuits A, B, C et D.

Une vue d'ensemble de l'architecture des circuits sélectionnés est représentée dans la figure 3.1 ci-dessous. Les circuits reliant les villes source et destination comprennent un ensemble de spans amplifiés. Il y'en a aussi des sites de ROADM permettant l'ajout et l'extraction de longueurs d'onde dans certaines villes parcourues. Le tableau 3.1 résume leurs caractéristiques. Parmi

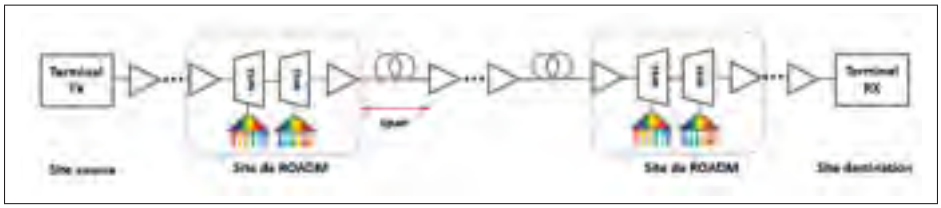


Figure 3.1 Architecture des circuits sélectionnés

ceux choisis, il y a un circuit aérien, un autre enfoui et deux autres dont le type d'installation est inconnu. Les circuits B et C appartiennent à la même route et sont bidirectionnels. Les deux autres circuits appartiennent à des routes différentes et éloignées géographiquement. Finalement, tous les circuits sont de type régional et ont des nombres d'observations comparables. La qualité

Tableau 3.1 Quelques caractéristiques pour les circuits sélectionnés

	Type d'installation	**Longueur (km)**	**Nombre de spans**	**Nombre d'observations**
Circuit A	aérienne	$\sim$ 545 km	16	33,443
Circuit B	inconnu	$\sim$ 424 km	3	33,475
Circuit C	inconnu	$\sim$ 424 km	3	33,797
Circuit D	enfouie	$\sim$ 600 km	27	33,463

de performance durant la période d'observation, traduite par le SNR, est représentée pour les quatre circuits sélectionnés dans la figure 3.2. Afin de visualiser la variation du SNR en plus de

détails, un *zoom* a été effectué sur le troisième mois. Notons que la qualité de performance pour le circuit B est semblable à celle du circuit C. Ceci est expliqué par la bidirectionnalité des deux circuits. Ensuite, la dispersion des valeurs de SNR observée pour chaque circuit est schématisée

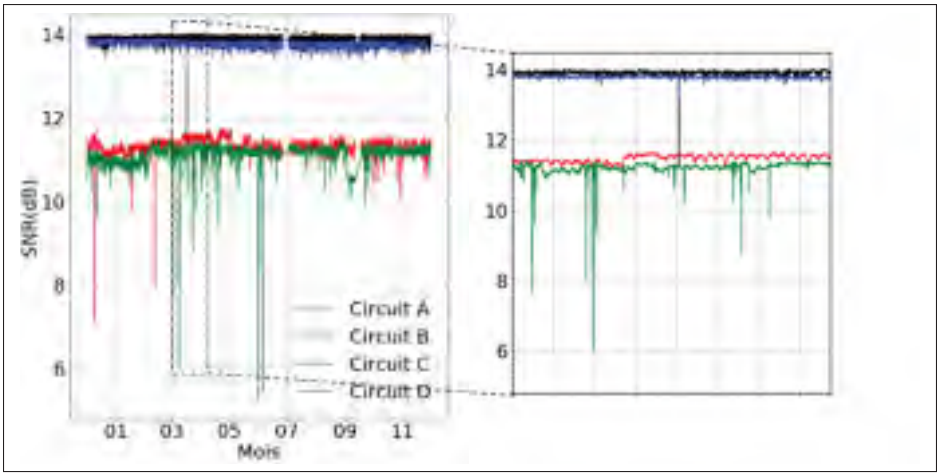


Figure 3.2 Représentation temporelle de la qualité de performance durant la période d'observation pour les circuits sélectionnés

dans la figure 3.3. La ligne horizontale (en violet) représente la médiane de la distribution des valeurs et les moustaches correspondent à 1.5 fois l'intervalle interquartile (*Interquartile Range* : IQR). L'IQR délimite le premier (noté $Q1$) et le troisième quartile (noté $Q3$). Les points représentés par des croix sont ceux qui dépassent la fin des moustaches et peuvent être considérés comme des valeurs aberrantes. Notons bien que les circuits A et D ont beaucoup plus d'apparitions de valeurs aberrantes durant cette période d'observation que pour les circuits C et D. Quelques statistiques descriptives sur la qualité de performance SNR sont données dans le tableau 3.2. Le coefficient de variation (*Coefficient of Variation* : CV) permet de comparer le degré de variation d'une observation à une autre. Il se calcule comme le ratio entre l'écart type et la moyenne des observations disponibles et s'exprime en pourcentage. Les circuits D et A respectivement ont plus de variation que les circuits B et C. La fonction d'autocorrélation (ACF) a aussi été implémentée. Si une périodicité est observée dans le corrélogramme, son amplitude (saisonnalité) est quantifiée en utilisant une décomposition additive de la série temporelle. La saisonnalité la plus prononcée appartient au circuit A et elle est de 24 heures. Le circuit B a à

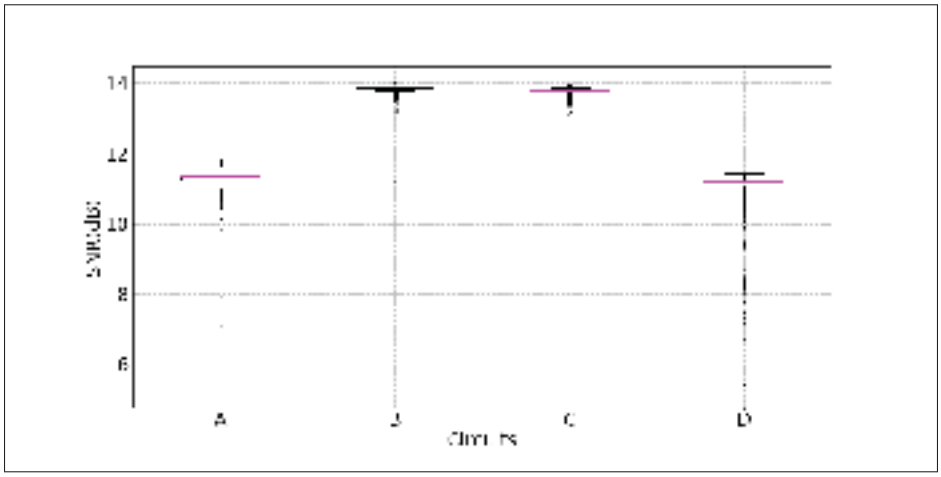


Figure 3.3 Représentation temporelle de la qualité de performance durant la période d'observation pour les circuits sélectionnés

son tour une faible saisonnalité de 12 heures. L'étude de la stationnarité effectuée sur les circuits

Tableau 3.2 Statistiques descriptives sur la qualité de performance des circuits sélectionnés

	SNR				
	Pourcentage de données manquantes (%)	**Moyenne (dB)**	**Médiane (dB)**	**CV (%)**	**Saisonnalité (dB)**
Circuit A	4.5576	11.35	11.35	0.012	0.08
Circuit B	4.4663	13.90	13.91	0.004	0.03
Circuit C	2.3293	13.79	13.78	0.003	0.02
Circuit D	4.5006	11.15	11.21	0.019	—

est résumée dans le tableau 3.3. Selon les résultats des tests statistiques ADF et KPSS (présentés dans la section 2.2.3.2), les quatre séries temporelles devraient être stationnarisées en effectuant une différenciation d'ordre 1.

Tableau 3.3 Étude de la stationnarité pour les circuits sélectionnés

	Test ADF	**Test KPSS**	**Ordre de la stationnarité**	**Comment stationnariser la série temporelle ?**
Circuit A	Stationnaire	Non-stationnaire	Ordre 1	Différencier la série à l'ordre 1
Circuit B	Stationnaire	Non-stationnaire	Ordre 1	Différencier la série à l'ordre 1
Circuit C	Stationnaire	Non-stationnaire	Ordre 1	Différencier la série à l'ordre 1
Circuit D	Stationnaire	Non-stationnaire	Ordre 1	Différencier la série à l'ordre 1

3.3 Traitement des données

Pour appliquer des techniques d'apprentissage automatique, il faudrait commencer par traiter et préparer les données. Les étapes indispensables concernent généralement l'imputation des données manquantes, la gestion des valeurs aberrantes et la normalisation des données. Chacune de ces étapes sera détaillée dans ce qui suit.

3.3.1 Imputation des données manquantes

La présence de valeurs manquantes dans les données est un problème assez courant surtout pour les applications du monde réel. Une attention bien particulière doit être accordée à l'imputation de ces données vu qu'elle pourra impacter négativement la performance des modèles prédictifs (Dunsmuir & Robinson, 1981). Ceci dépend nécessairement de leurs fréquences et temps d'apparition. Le pourcentage des données manquantes a été quantifié pour chacun des circuits (tableau 3.2). Le circuit B possède beaucoup plus de données manquantes (4.47%) comparativement au circuit C (2.33%). La fréquence d'apparition des données manquantes pour chaque circuit étudié est illustrée à la figure 3.4. Notons que la présence de données manquantes à la fin du sixième mois et au début du septième (environ 5 jours) ainsi que pendant le huitième mois (environ 3 jours) touche à tous les circuits malgré qu'ils soient répartis géographiquement.

Cela est possiblement dû à une panne au niveau du système de collecte de données. Généralement, la perte de données se produit sur plusieurs journées consécutives formant ainsi des creux. Plusieurs techniques peuvent être utilisées pour remédier à ce problème :

- **suppression des données manquantes :** elle est utilisée sous condition que les valeurs manquantes ne soient pas si nombreuses ou n'aient pas un grand effet sur le comportement global de la série temporelle. Cette solution sera considérée si la fréquence d'apparition de données manquantes par mois ne dépasse pas 10% de l'ensemble de données manquantes du circuit. Cette valeur est choisie de façon expérimentale. En effet, en supprimant au-delà de ce seuil, les performances des modèles prédictifs ont été impactées pour les quatre circuits étudiés. Par exemple, cette technique sera appliquée pour les circuits A et C pendant les trois premiers mois ;
- **imputation des données manquantes avec les modèles d'apprentissage** : ceci consiste à prédire les valeurs des données manquantes et les remplacer dans la série temporelle d'origine. Un réseau de neurones de type LSTM sera utilisé dans notre cas pour imputer les données manquantes dont la fréquence par mois est plus que 10% de l'ensemble de données manquantes du circuit ;
- **remplacement des valeurs manquantes :** ceci peut se faire en remplaçant la valeur manquante par la moyenne ou par la médiane ou encore par la dernière valeur connue de la série temporelle. Ayant des séries non stationnaires, la moyenne peut fluctuer considérablement pour différentes parties de la série. Ainsi, cette technique n'est pas applicable dans notre cas.

3.3.2 Gestion des valeurs aberrantes

L'existence des valeurs aberrantes dans les séries temporelles est un problème inévitable. Ces dernières peuvent diminuer la performance des modèles prédictifs. Par conséquent, il est important de les détecter et de prendre les mesures nécessaires par rapport à ces points anormaux dans l'étape de prétraitement des données. Dans ce projet, les données aberrantes constituent un ensemble minoritaire d'observations. Elles ont aussi des valeurs qui s'écartent énormément des données normales. Dans le cas des réseaux optiques opérationnels, les valeurs aberrantes

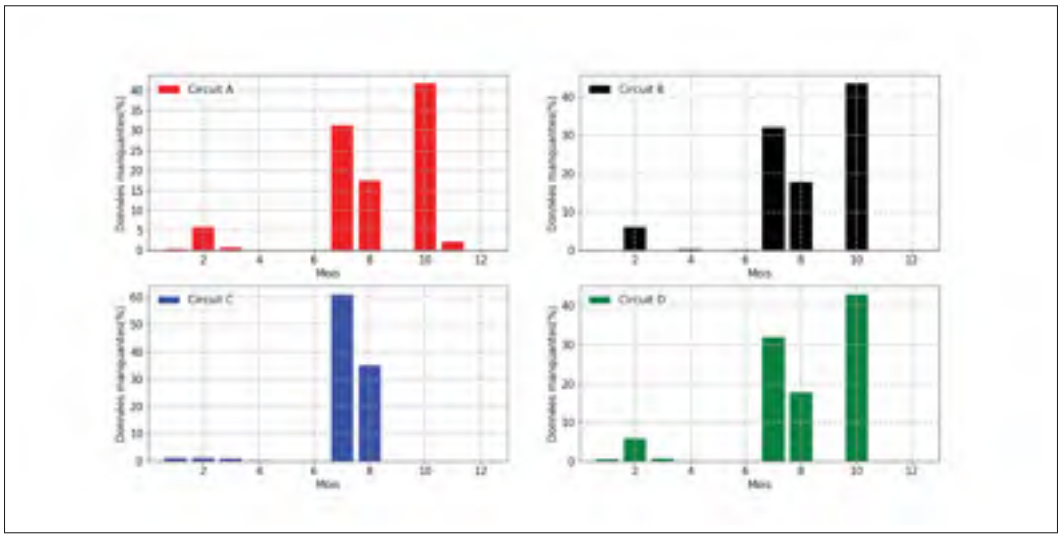


Figure 3.4 Répartition des données manquantes sur les mois d'observations pour les circuits sélectionnés

devraient survenir pendant un dysfonctionnement du réseau ou pendant une réparation ou opération de maintenance (e.g., pincement d'un câble de fibre optique par un opérateur). Étant donné que les opérations de maintenance sont généralement rapides, la durée des observations aberrantes ne devrait pas dépasser quelques heures. En somme, les valeurs aberrantes sont peu fréquentes, différentes des autres données et d'une durée limitée. Une fois détectés, ces points peuvent être supprimés de la série temporelle d'origine ou imputés par d'autres valeurs. La détection se fait en trois étapes : d'abord, développer un modèle capable de reconnaitre le comportement normal de la série temporelle. Ensuite, déterminer une limite de décision (‘*decision boundry*’) permettant de classifier une observation en tant qu'une valeur normale ou aberrante. Finalement, décider si cette limite est significative ou non. Les algorithmes de détection d'anomalies sont appropriés surtout pour des données ayant une distribution normale. Cependant, leurs résultats restent très acceptables pour d'autres distributions. Dans notre cas, les valeurs aberrantes sont traitées uniquement pour les données de performance (SNR). En effet, un autre projet portant sur les anomalies est en cours au sein de notre équipe. Avant d'aborder la détection des valeurs aberrantes, il est intéressant de tracer l'histogramme des données afin de mieux comprendre leurs distributions. Ce dernier permet de visualiser la fréquence des données qui se situent dans une plage de valeurs spécifiées. Comme le montre la figure 3.5, la distribution

du circuit C est similaire à une loi normale. Cependant, ce n'est pas tout à fait le cas pour le reste des circuits.

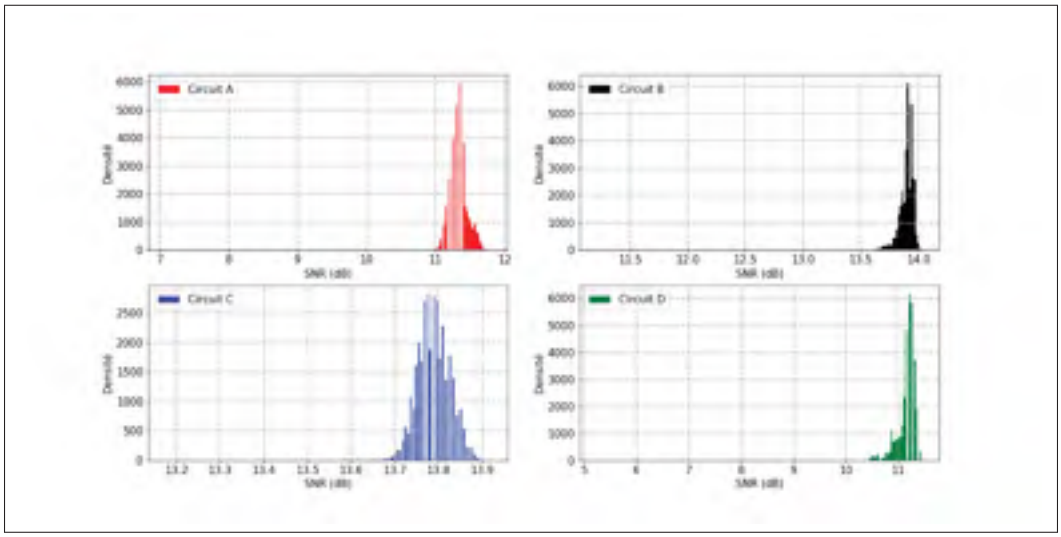


Figure 3.5 Histogrammes des SNRs pour les circuits sélectionnés

Plusieurs algorithmes de détection d'anomalies ont été testés durant ce projet. La technique de mélange de gaussienne avec k composantes est présentée dans ce rapport. Il s'agit d'une méthode d'estimation de densité où le but est de trouver la loi de probabilité $p(x)$ qui a généré les données. Un mélange de gaussienne suppose que les données sont échantillonnées à partir de k différentes lois gaussiennes ayant chacune des moyennes et covariances distinctes. Les paramètres du modèle (moyennes et covariances) sont estimés sur l'ensemble d'entraînement. Afin de déterminer le nombre de composantes (correspondant au nombre de classes ou *clusters*) optimal pour chaque circuit, le coefficient de silhouette a été mesuré sur l'ensemble de validation. Ce coefficient fournit une mesure de la qualité du partitionnement d'un échantillon dans une classe donnée en fonction de la distance séparant les classes et de leurs étanchéités. Il est calculé pour chaque échantillon disponible. Cette analyse permet d'évaluer la distance de séparation entre les différentes classes. Il est calculé comme suit :

$$\text{Coefficient}_{silhouette} = \frac{b - a}{\max(a, b)} \tag{3.1}$$

Où :

a et *b* représentent respectivement la distance moyenne avec les échantillons de sa classe et la distance entre l'échantillon considéré et la classe la plus proche de celle auqelle il appartient.

Ce coefficient prend des valeurs entre ±1 . Une valeur de 1 indique que les données dans une classe sont loin des autres classes. Cependant, un coefficient de silhouette proche de 0 signifie que la définition d'une limite de décision entre les classes voisines n'est pas aussi pertinente. Des valeurs négatives dévoilent que les échantillons sont peut-être mal partitionnés. Cette analyse a été faite pour différent nombre de composantes. Un exemple des résultats obtenus pour un nombre de composantes égal à 2 est illustré dans la figure 3.6. Les coefficients de silhouette moyens mesurés pour les circuits A, B, C et D sont respectivement de : 0.41, 0.66, 0.78 et 0.78. Les répartitions des données ainsi que les tailles des classes sont aussi visualisées sur la même figure. Un nombre de composantes égal à 2 s'avère un mauvais choix pour le circuit C en raison de la présence d'une classe avec un coefficient de silhouette inférieur à la moyenne. Pour les circuits A, B, C et D respectivement, les nombres de composantes retenus sont : 3, 3, 2 et 3. Le

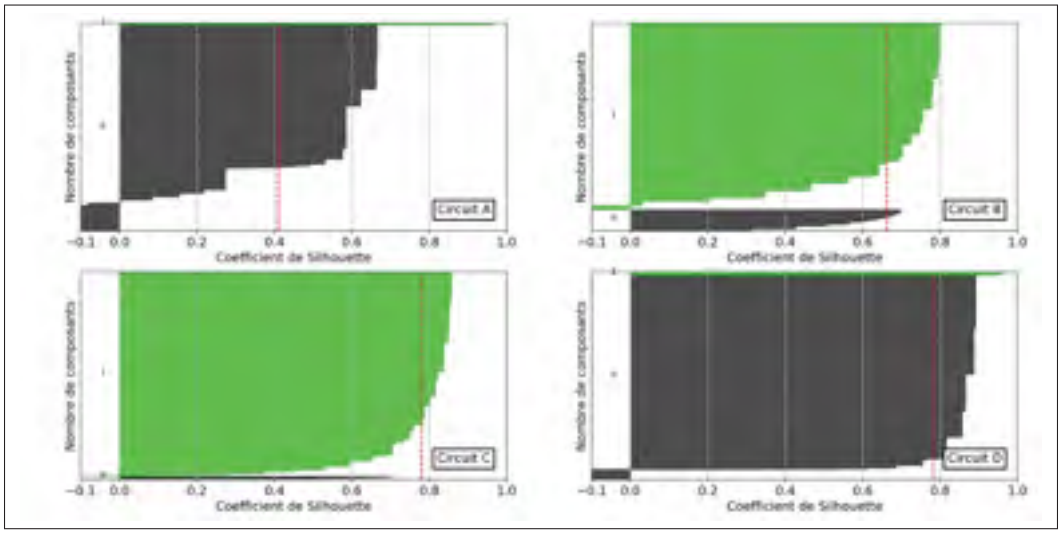


Figure 3.6 Représentation des coefficients de silhouette en fonction du nombre de composants pour les circuits sélectionnés

logarithme de la probabilité marginale est calculé pour tous les circuits étudiés et représenté dans la figure 3.7. En ayant des exemples étiquetés de valeurs aberrantes, le seuil est défini de

façon à minimiser le taux de faux positifs sur l'ensemble de validation. Dans notre cas, il n'y a pas de données étiquetées. Ainsi, le seuil est déterminé de façon visuelle.

Les valeurs aberrantes qui ont été identifiées par cette méthode sont celles représentées en noir

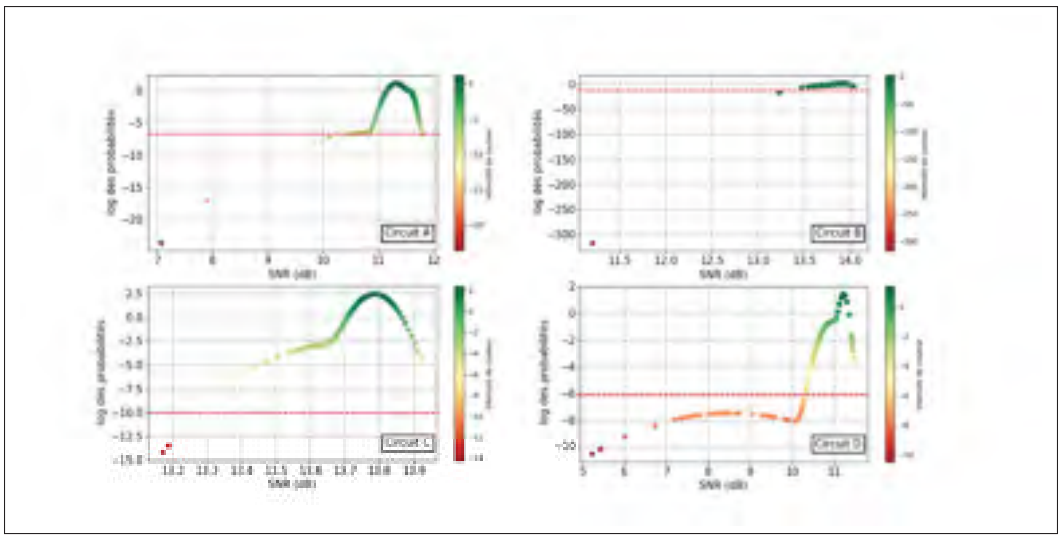


Figure 3.7 Probabilités marginales pour les circuits étudiés

dans la figure 3.8. Le nombre de valeurs aberrantes ayant été détecté pour les circuits A, B, C et D sont respectivement de : 41, 3, 2 et 67. Quant aux durées maximales des anomalies adjacentes, elles sont égales à : 4 heures :45 minutes pour le circuit A, 45 minutes pour circuit B, 30 minutes pour circuit C et finalement 5 heures :30 minutes pour circuit D. Le nombre et durée de valeurs aberrantes mentionnées valide l'hypothèse que les anomalies sont peu fréquentes, différentes des autres données et d'une durée limitée. Selon le type de scénario évalué, les valeurs aberrantes seront soit gardées, soit éliminées ou imputées en utilisant un réseau de type LSTM.

3.3.3 Normalisation des données

La non-normalisation des données à l'entrée des réseaux de neurones peut augmenter considérablement le temps d'entraînement et peut aussi influencer négativement les performances des modèles de prédiction (Hochreiter & Schmidhuber, 1997). Si la série temporelle est multivariée,

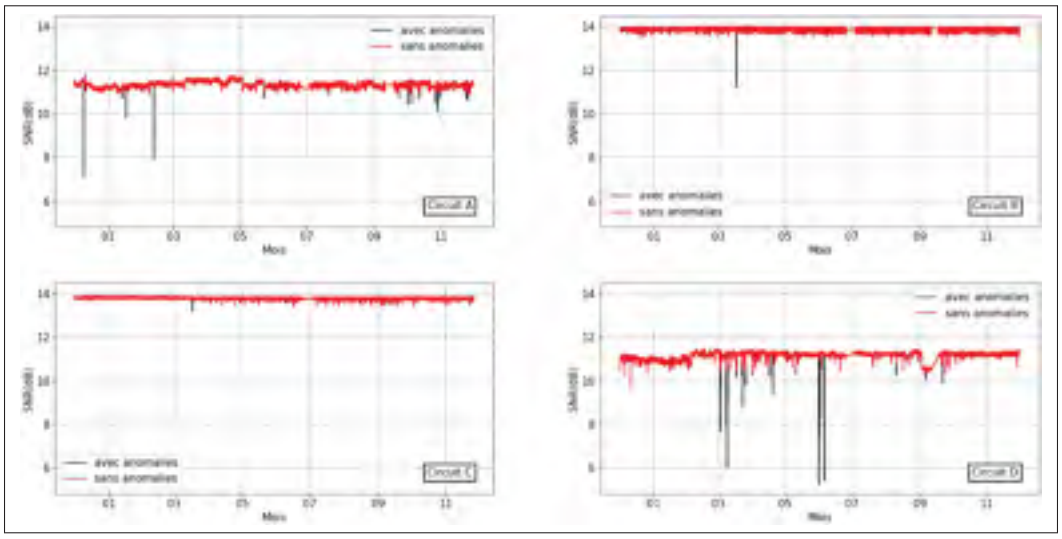


Figure 3.8 Identification des anomalies pour les circuits étudiés

chaque attribut sera normalisé indépendamment des autres. Deux techniques de normalisation (dite aussi mise à l'échelle) ont été appliquées dans le cadre de ce projet.

- la première consiste à projeter l'ensemble des valeurs prises par la série temporelle sur un intervalle donné situé entre un minimum et un maximum (noté [min,max]). Par exemple, les données peuvent être projetées sur l'intervalle [-1,1], ou encore [0,1]. La transformation est représentée par l'équation suivante :

$$y_i^{\text{transformé}} = \frac{y_i - y_{min}}{y_{max} - y_{min}} \tag{3.2}$$

  Où :

  y_i, y_{max} et y_{min} représentent respectivement la valeur prise par la série à l'instant $t = i$ et ces valeurs minimales et maximales. Cette technique très sensible à la présence de valeurs aberrantes sera testée uniquement pour les circuits B et C ;

- la deuxième technique n'est pas si différente de la première, mais elle est robuste à la présence des valeurs aberrantes. En effet, la transformation se fait en soustrayant la médiane de la série

temporelle et en divisant par l'intervalle interquartile (3.3).

$$y_i^{\text{transformé}} = \frac{y_i - \text{mediane}(y)}{Q_3(y) - Q_1(y)} \tag{3.3}$$

Où :

mediane(y), $Q_3(y)$ et $Q_1(y)$ représentent respectivement la médiane de la série temporelle, son troisième et premier quartile.

Cette technique sera principalement considérée pour les circuits A et D.

3.4 Sélection des attributs explicatifs pour la prédiction multivariée

Identifier les attributs pertinents pour la prédiction est une étape indispensable pour avoir des modèles performants. Plusieurs avantages découlent de cette démarche. Premièrement, une base de données d'entraînement contenant un nombre élevé d'attributs peut causer le surapprentissage. En effet, en ayant plusieurs attributs qui n'influencent pas nécessairement la variable prédite, le modèle peut capturer des effets aléatoires. Deuxièmement, entraîner un modèle possédant une dimensionnalité réduite nécessite moins de ressources et moins de temps d'exécution. Finalement, un modèle exploitant beaucoup d'attributs fortement corrélés est complexe. Il est ainsi difficilement interprétable. L'étude de la corrélation est généralement la première étape pour sélectionner les attributs explicatifs de la variable prédite. Le coefficient de corrélation mesure le degré d'association entre deux variables, autrement dit leur tendance à changer ensemble. Il décrit aussi la direction de ce changement. Ce coefficient prend des valeurs entre ± 1. Une valeur de ± 1 indique une association parfaite entre les variables. Le sens du changement est donné par le signe de ce dernier coefficient. Les corrélations de *Pearson* et *Spearman* ont été utilisées dans cette analyse. Le premier type de corrélation identifie une association linéaire entre deux attributs. Quand deux variables augmentent ou diminuent simultanément avec un taux constant, une relation linéaire positive existe entre elles. La deuxième permet d'identifier une relation monotone entre les variables. Dans une relation monotone, les variables ont tendance à se déplacer dans la même direction avec un taux qui n'est pas nécessairement

constant. Un coefficient de *Spearman* nul n'implique pas l'absence totale d'association entre les deux variables, mais plutôt l'absence de relation monotone.

Les résultats d'analyse seront illustrés uniquement pour le circuit A. Le même processus peut être refait pour le reste des circuits. Notons qu'il n'y a aucune garantie d'obtenir exactement les mêmes résultats et conclusions. Les attributs explicatifs étudiés dans ce stade du projet sont :

- **DGD_MAX :** délai de groupe différentiel maximal enregistré sur un intervalle de 15 minutes ;
- **OPR_MAX :** puissance maximale du signal optique reçu au bout de 15 minutes ;
- **Pre_FEC :** Nombre d'erreurs corrigées par le FEC pendant chaque seconde. Ce compteur d'erreurs continue à incrémenter jusqu'à atteindre un seuil par seconde défini par l'opérateur ;
- **OPT_MAX :** puissance maximale du signal optique transmise au bout de 15 minutes ;
- **HCCS :** en anglais *High Correction Count Second*. C'est le nombre de secondes par intervalle de 15 minutes où le compteur Pre_FEC a dépassé sa valeur seuil ;
- **ORL_MAX :** Perte maximale de puissance du signal résultant de la réflexion sur un intervalle de 15 minutes. Cette valeur est monitorée sur tous les amplificateurs de lignes.

Les valeurs maximales ont été choisies dans cette analyse, car ce sont elles qui perturbent le plus la qualité de performance.

Le circuit A contient 16 amplificateurs de ligne. Ainsi, l'ORL_MAX n'a pas été représenté sur les matrices de corrélation pour ne les pas encombrer. Il est à noter que l'association entre les différentes valeurs d'ORL_MAX et la variable cible SNR ne dépasse pas le 0.1.
D'après les matrices de corrélation de *Pearson* et *Spearman* en bas, il existe une relation linéaire parfaite et décroissante entre les variables Pre_FEC et SNR. Quant aux variables OPR_MAX et SNR, elles sont fortement reliées avec une corrélation positive. Théoriquement, ces deux derniers résultats sont attendus. Finalement, les corrélations entre OPT_MAX, DGD_MAX et HCCS et la variable prédite SNR sont très faibles ou nulles. L'absence de forte relation entre ORL_MAX, OPT_MAX, DGD_MAX avec le SNR est attendue. En effet, un réseau optique cohérent compense à priori ses dégradations. Au meilleur de nos connaissances, il n'y a pas eu

de recherches qui étudient la relation entre HCCS et le SNR. Notons aussi que la corrélation entre Pre_FEC et OPR_MAX est élevée. Dans les figures ci-dessous, les relations entre certains des attributs explicatifs et la variable prédite sont schématisées.

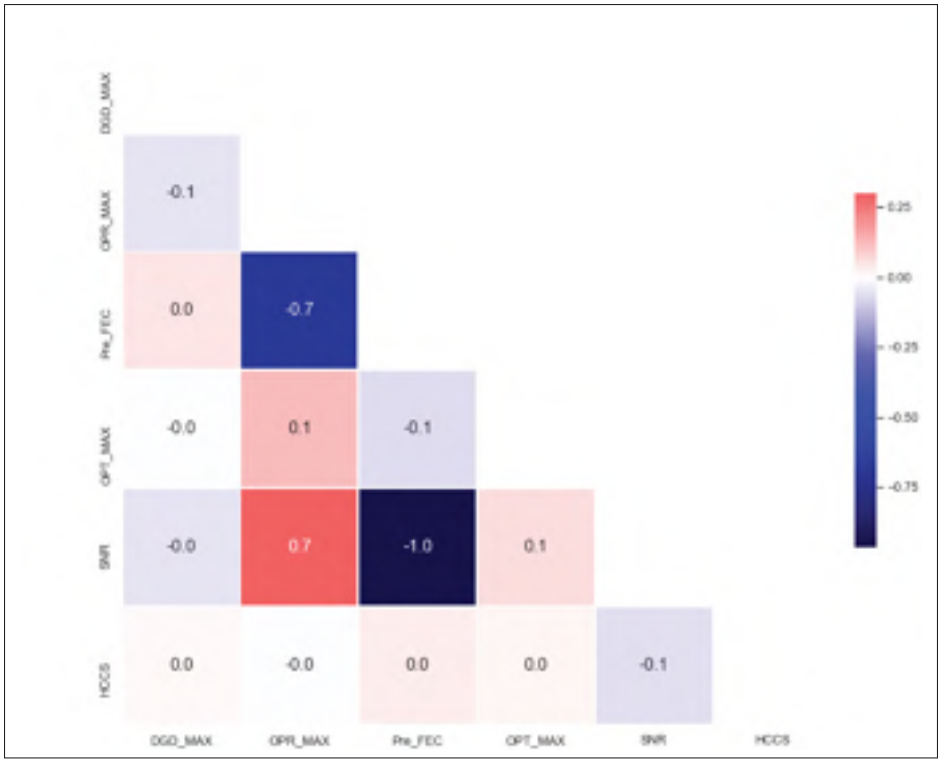


Figure 3.9 Corrélation de Pearson pour les différents attributs étudiés, circuit A

Dans un modèle idéal, il est intéressant d'avoir des attributs explicatifs complètement indépendants entre eux, mais qui sont corrélés avec la variable prédite. L'analyse de multi colinéarité et la mesure d'importance des variables explicatives présentées dans les deux sections suivantes sont les pratiques couramment utilisées afin de sélectionner les attributs pertinents pour la prédiction temporelle.

3.4.1 Analyse de multi colinéarité

La multi colinéarité, un terme largement employé en statistiques, réfère à l'utilisation du même type information plusieurs fois dans un modèle de prédiction. Dans notre cas, elle se produit

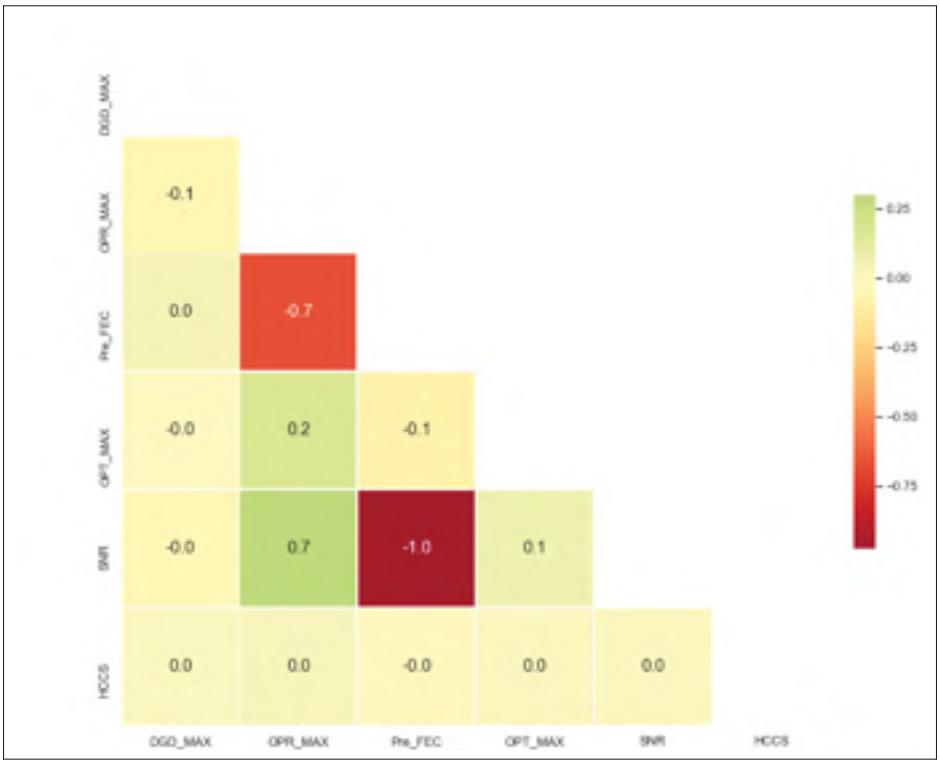


Figure 3.10 Corrélation de Spearman pour les différents attributs étudiés, circuit A

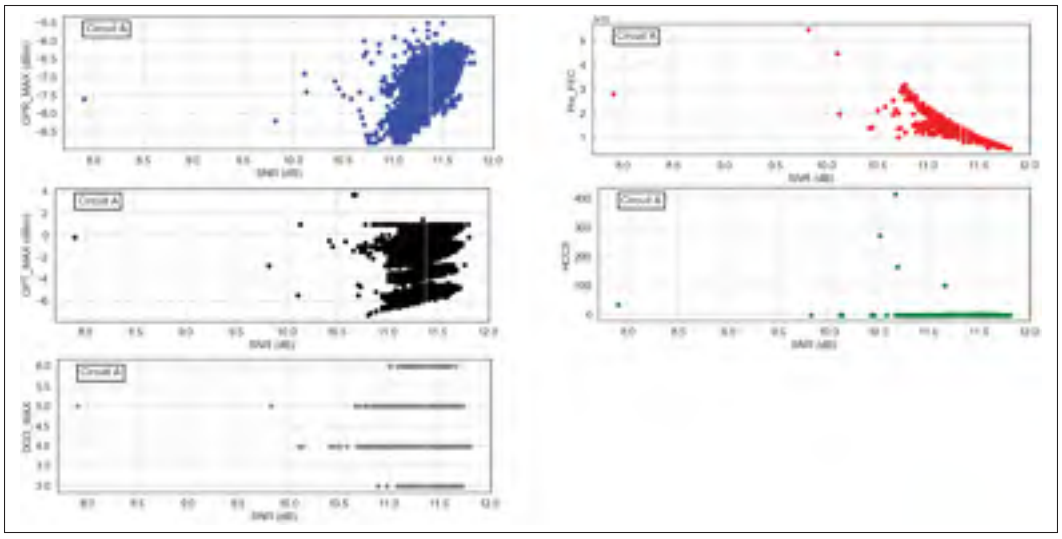


Figure 3.11 Relations entre SNR et les attributs explicatifs, circuit A

quand plusieurs (plus que deux) attributs explicatifs introduits dans le modèle sont fortement corrélés entre eux. La présence de multi colinéarité n'influence pas nécessairement la précision du modèle de prédiction. Cependant, elle agit sur la sensibilité de ses coefficients : un petit changement dans le modèle peut résulter en une large fluctuation des coefficients. La multi colinéarité réduit de la même façon la précision des coefficients ce qui affaiblit la crédibilité statistique du modèle. La nécessité de gérer la multi colinéarité dépend de son type et de sa sévérité. Deux types de multi colinéarité existent. La première est la multi colinéarité structurelle. Pour la traiter, il faudrait éliminer les attributs dérivés d'autres attributs tels que le SNR, BER et facteur Q. Le deuxième type est la multi colinéarité venant des données en elles-mêmes. Une des techniques utilisées pour quantifier la sévérité de ce type est le calcul des facteurs d'inflation de la variance (*Varaiance Inflation Factor* : VIF). Ce dernier facteur mesure l'impact de la multi colinéarité dans l'augmentation des variances des coefficients du modèle. Il permet de connaitre le degré auquel un attribut peut être expliqué par les autres attributs du modèle. Le VIF a des valeurs supérieures ou égales à 1. Si le VIF est égal à 1, l'attribut testé n'est pas multi colinéaire avec d'autres attributs. Quand la valeur est comprise entre 1 et 5, la multi colinéarité est modérée. Un VIF supérieur à 5 représente un cas sévère de multi colinéarités. Ceux possédant des valeurs élevées seront retirés de la liste des attributs explicatifs. Pour le circuit A, seule la multi colinéarité structurelle existe. Cependant, la multi colinéarité venant des données a été traitée pour d'autres circuits étudiés.

Le Pre_FEC et OPR_MAX sont fortement corrélés. Dans ce cas précis, l'attribut qui porte le plus d'information devra être utilisé alors que l'autre sera éliminé. La mesure de l'importance des attributs explicatifs peut se faire à l'aide de la permutation.

3.4.2 Mesure de l'importance des variables explicatives

L'importance d'un attribut explicatif peut être déterminée en fonction de sa contribution dans l'amélioration ou dégradation de la performance d'un modèle prédictif. Une des techniques consiste à permuter aléatoirement l'ordre des échantillons d'une variable explicative et mesurer l'impact de cette opération dans les résultats de performance. Cette procédure permet de

rompre toute relation (chronologique dans le cas des séries temporelles) entre la variable prédite et l'attribut explicatif. Si ce dernier est d'une importance majeure pour la prédiction, les performances seront dégradées. L'avantage de cette technique est qu'elle est indépendante du modèle de prédiction. En appliquant cette technique de permutation avec un réseau de neurones de type LSTM, c'est le OPR_MAX qui apporte le plus d'information pour le circuit A.

3.5 Mesure de la performance de la qualité de prédiction

Cinq métriques ont été considérées pour évaluer la performance des modèles de prédiction. Chacune d'elles permet d'interpréter la qualité du modèle d'un point de vue différent.

Biais :

Le biais représente la moyenne de l'erreur de prédiction. Il est exprimée par :

$$Biais = \frac{\sum_{t=1}^{N}(y_t - \hat{y}_t)}{N} \tag{3.4}$$

Où :

y_t et $\hat{y}_t$ et N représentent respectivement la valeur réelle et prédite à l'instant t et le nombre d'observations au total.

Cette mesure permet d'évaluer la direction moyenne de l'erreur. Si le modèle de prédiction surestime la qualité de performance, le biais sera positif. Dans le cas contraire, il aura une valeur moyenne négative. L'idéal serait d'avoir un biais nul. Dans notre cas, il est parfois dangereux de surestimer la qualité de performance. En effet, si l'opérateur désire emprunter de la marge pour une connexion en panne, il peut baser sa décision sur la qualité de performance prédite pour la prochaine journée. Ceci lui permet de déterminer la marge de SNR qui lui est disponible dans le futur proche. Une large surestimation de la qualité de performance peut ainsi entraîner des imprévus ou même nuire aux accords de niveau de service.

Erreur absolue moyenne :

Cette métrique abréviée dans ce qui suit par MAE (*Mean Absolute Error*) mesure la moyenne

de la valeur absolue de l'erreur dans la même unité que les données d'origine. L'expression mathématique du MAE est la suivante :

$$MAE = \frac{\sum_{t=1}^{N} |y_t - \hat{y}_t|}{N} \tag{3.5}$$

Un MAE nul indique que les erreurs de prédiction sont nulles pour tous les exemples évalués. L'utilisation de la valeur absolue empêche les erreurs négatives et positives de s'annuler mutuellement. La principale propriété du MAE est qu'elle donne le même poids aux erreurs, qu'elles soient larges ou petites. Par exemple, une erreur de '2' a le même effet que deux erreurs de '1'.

Racine carrée de l'erreur quadratique moyenne :

La racine carrée de l'erreur quadratique moyenne, notée RMSE (*Root Mean Squared Error*) mesure l'écart type de l'erreur de prédiction. Cette métrique est exprimée par :

$$RMSE = \sqrt{\frac{\sum_{t=1}^{N} (y_t - \hat{y}_t)^2}{N}} \tag{3.6}$$

Le RMSE a la même unité de mesure que les données d'origine, soit dans notre cas le dB. Plus le RMSE est petit, plus le modèle de prédiction est précis, et vice versa. L'élèvement au carré des valeurs empêche les erreurs négatives et positives de se compenser entre elles. De plus, cette opération pénalise lourdement les erreurs larges. Cette dernière propriété a à la fois un avantage et un inconvénient. En effet, une grande variation dans les erreurs de prédiction aura tendance à augmenter de manière très prononcée le RMSE, ce qui n'est pas souhaitable. Cette métrique favorise plutôt les modèles de prédiction produisant des erreurs d'une ampleur cohérente à chaque instant. Du côté négatif, la qualité d'un modèle qui a de bonnes prédictions partout sauf en quelques points peu nombreux est masquée. Du coup, le RMSE ne peut pas être utilisé seul pour juger la précision des modèles de prédiction.

Le MAE et RMSE présentés ci-dessus permettent de comparer l'efficacité de différents modèles de prédiction lorsqu'elles sont appliquées à la même série temporelle. Cependant, elles sont

moins utiles quand il est question de comparer la performance d'un modèle sur plusieurs séries temporelles surtout s'ils ont des échelles de valeurs différentes.

Coefficient de détermination :

Le coefficient de détermination notée R^2 représente la proportion de variance dans les observations réelles (y) qui a pu être expliquée par les attributs d'entrée du modèle. Il fournit ainsi une vue sur la qualité de prédiction en ayant de nouvelles données de test. Cette métrique est donnée par l'équation suivante :

$$R^2(y, \hat{y}) = 1 - \frac{\sum_{t=1}^{N} (y_t - \hat{y}_t)^2}{\sum_{t=1}^{N} (y_t - \bar{y}_t)^2} \tag{3.7}$$

Où :

$$\bar{y}_t = \sum_{t=1}^{N} \frac{y_t}{N} \tag{3.8}$$

Si les prédictions sont proches des valeurs réelles, R^2 sera proche de 1. Dans le cas contraire, R^2 sera très proche de 0. Cette métrique peut aussi avoir des valeurs négatives indiquant une qualité de prédiction médiocre. Prenons l'exemple d'un $R^2 = 0.1$, dans ce cas de figure 90% de la variance dans la variable réelle ne peut pas être expliqué par le modèle. Il n'y a pas de recommandations par rapport à la valeur de R^2 qu'on devrait avoir pour juger la qualité du modèle. Tout dépend de l'application. Lorsque de nombreux facteurs incontrôlables, indéterminables et inconnus influencent la variable réelle, une valeur de R^2 proche de 0.1 peut être très appréciable. Le coefficient de détermination est utilisé pour comparer la qualité de plusieurs modèles de prédiction sur une même série temporelle. Un des inconvénients de cette métrique est qu'elle continue à augmenter en ajoutant des attributs d'entrées au modèle, même s'ils ne sont pas tant explicatifs et pertinents pour la prédiction. Dans ce cas, le risque de surapprentissage est majeur. Le modèle pourra produire des valeurs de R^2 trompeusement élevées alors que la précision de la prédiction est réduite.

Coefficient de détermination ajusté :

Le coefficient de détermination ajusté est utilisé pour comparer deux modèles de prédiction ayant un nombre différent d'attributs explicatifs. La valeur de cette métrique augmente uniquement

dans le cas où l'ajout d'un autre attribut pour le modèle améliore réellement la prédiction. Cependant, elle diminue si son ajout n'est pas aussi bénéfique. Le coefficient de détermination ajustée est donné par l'équation 3.9. Sa valeur est inférieure ou égale à celle du R^2.

$$R^2_{\text{ajusté}} = 1 - \left[\frac{\left(1 - R^2\right)(N-1)}{N-K-1} \right] \tag{3.9}$$

Où :

N et K représentent respectivement le nombre d'échantillons de données et le nombre d'attributs explicatifs introduits dans le modèle de prédiction.

3.6 Modèles de prédiction étudiés

La prédiction de la qualité de performance pour les réseaux optiques opérationnels se décrit comme un problème d'apprentissage automatique supervisé de type régression. Nombreuses sont les techniques d'apprentissage automatique et d'apprentissage profond pouvant répondre à cette problématique. N'ayant pas d'études précédentes qui ont traité le même sujet, nos choix de modèles ont été basés sur des domaines connexes tels que la finance ou l'énergie renouvelable. Les réseaux de neurones de type LSTM, les réseaux de neurones de type GRU et les réseaux de neurones 1D-CNN sont les méthodes les plus populaires dans la littérature et qui seront évaluées dans ce stade du projet. Pour la prédiction temporelle univariée, les modèles mentionnés seront comparés avec les méthodes de référence (baselines) ARIMA et SARIMA dépendamment de l'existence ou non d'une saisonnalité dans les données des circuits. Les hyperparamètres de chacune des méthodes sont présentés dans ce qui suit. Les modèles de prédiction prennent en entrées un vecteur contenant une fenêtre historique du SNR et prédit les prochaines 24 heures de SNR (figure 3.12).

ARIMA :

L'idée principale de la méthode ARIMA est de prédire les valeurs futures en se basant uniquement sur les informations capturées dans les valeurs passées de la série temporelle.

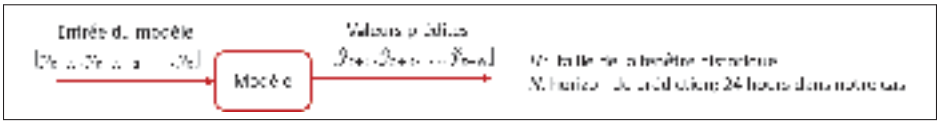


Figure 3.12 Processus de la prédiction à l'instant t

Toute série temporelle non saisonnière et qui n'est pas un simple bruit blanc aléatoire peut être modélisée à l'aide d'un modèle ARIMA. Les paramètres caractérisant un modèle ARIMA sont :

- p : ordre du terme autorégressif (AR) ;
- q : ordre du terme à moyenne mobile (MA) ;
- d : ordre de différentiation pour obtenir une série temporelle stationnaire.

SARIMA :

SARIMA est un modèle ARIMA capable de gérer la composante saisonnière dans la série temporelle. Les hypothèses de ce modèle sont les suivantes : premièrement, le modèle SARIMA suppose l'existence d'une corrélation linéaire entre les échantillons de données au présent et aux instants passés de la variable prédite. Deuxièmement, il suppose l'existence d'un bruit aléatoire aux instants présents et passés affectant la variable prédite. Troisièmement, cette méthode admet l'absence d'effet significatif d'autres variables exogènes dans la série temporelle sur la variable prédite.

Les paramètres p, q et d sont partagés avec le modèle ARIMA. Ceux spécifiques au SARIMA sont :

- P : ordre du terme autorégressif saisonnier (SAR) ;
- Q : ordre du terme à moyenne mobile saisonnier (SMA) ;
- D : ordre de différentiation saisonnière ;
- m : nombre de périodes dans chaque saison pour la variable prédite.

LSTM :

Le réseau de neurones récurrent de type LSTM permet de pallier les faiblesses des réseaux récurrents. Il est capable d'apprendre les dépendances à long terme dans les données. Une unité LSTM est généralement composée d'une cellule, d'une porte d'entrée, d'une porte de sortie et d'une autre d'oubli. Ces portes permettent de réguler le flux d'informations entrant et sortant de la cellule. Les configurations particulières *stateless* et *stateful* du réseau LSTM ont été évaluées dans ce projet. En utilisant la première configuration. La cellule mémoire est initialisée à zéro après chaque batch (la notion de batch est expliquée dans la section 3.7.2). Brièvement, les données sont passées au réseau de neurones sous forme de batch autrement en lots de données ayant une taille fixe. Dans ce cas de figure, le '*batch size*' est un hyperparamètre de grande importance. La configuration *stateless* est convenable pour les séries temporelles ayant une faible dépendance entre ces batchs. Cependant, dans la configuration *stateful*, l'état de la cellule mémoire est maintenu sur toute l'époque d'entraînement : en passant d'un batch à un autre, la cellule LSTM est initialisée au contenu du batch précédent. Cette dernière configuration requiert un temps d'entraînement plus élevé que la première.

Il y'en a des hyperparamètres qui sont commus entre les modèles LSTM, GRU et 1D-CNN. Ces derniers sont :

- le *batch size*, noté n_{batch} ;
- le nombre d'époques, noté n_{epochs} ;
- le taux d'apprentissage initial, noté $\alpha_{initial}$;
- la taille de la fenêtre historique des données, notée H.

Les hyperparamètres du modèle LSTM sont résumés dans le tableau 3.4. Chacun de cette liste sera pris avec beaucoup plus de détails dans les sections qui suivent.

GRU :

Un réseau récurrent GRU est une variation simple du LSTM. Tout comme le LSTM, le GRU est capable de conserver la mémoire pendant une longue durée. Ayant moins de paramètres, ce type de réseau a l'avantage d'être beaucoup moins coûteux en termes de ressources et en termes de

Tableau 3.4 Hyperparamètres du modèle LSTM

Symbole	Hyperparamètre
L_{LSTM}	Nombre de couches cachées
$\mathbf{N}_{LSTM}$	Vecteur contenant le nombre de neurones LSTM dans chaque couche cachée, $\mathbf{N}_{LSTM} = \left[n_{1_{LSTM}}, n_{2_{LSTM}}, \ldots, n_{l_{LSTM}}, \ldots, n_{L_{LSTM}}\right]$
$\mathbf{A}_{LSTM}$	Vecteur contenant les fonctions d'activation dans chaque couche cachée, $\mathbf{A}_{LSTM} = \left[a_{1_{LSTM}}, a_{2_{LSTM}}, \ldots, a_{l_{LSTM}}, \ldots, a_{L_{LSTM}}\right]$
$\mathbf{A}^{rec}_{LSTM}$	Vecteur contenant les fonctions d'activation récurrentes dans chaque couche cachée, $\mathbf{A}_{LSTM} = \left[a_{1^{rec}_{LSTM}}, a_{2^{rec}_{LSTM}}, \ldots, a_{l^{rec}_{LSTM}}, \ldots, a_{L^{rec}_{LSTM}}\right]$
$\mathbf{R}_{LSTM}$	Vecteur contenant les taux de *dropout* dans chaque couche cachée, $\mathbf{R}_{LSTM} = \left[r_{1_{LSTM}}, r_{2_{LSTM}}, \ldots, r_{l_{LSTM}}, \ldots, r_{L_{LSTM}}\right]$
$\mathbf{R}^{rec}_{LSTM}$	Vecteur contenant les taux de *recurrent dropout* dans chaque couche cachée, $\mathbf{R}_{LSTM} = \left[r_{1^{rec}_{LSTM}}, r_{2^{rec}_{LSTM}}, \ldots, r_{l^{rec}_{LSTM}}, \ldots, r_{L^{rec}_{LSTM}}\right]$

temps d'exécution. En effet, seuls deux types de portes sont pris en compte par le réseau GRU. Il s'agit d'une porte de pertinence et d'une autre de modification. L'unité GRU contrôle le flux d'informations similairement à l'unité LSTM, mais sans avoir recours à une unité de mémoire. Les mêmes configurations *stateful* et *stateless* existent pour le réseaux GRU. En pratique, les résultats de performance d'un modèle GRU sont comparables au modèle LSTM. Le réseau de neurones GRU partage quasiment les mêmes hyperparamètres que le LSTM (tableau 3.5).

Tableau 3.5 Hyperparamètres du modèle GRU

Symbole	Hyperparamètre
L_{GRU}	Nombre de couches cachées
$\mathbf{N}_{GRU}$	Vecteur contenant le nombre de neurones GRU dans chaque couche cachée, $\mathbf{N}_{GRU} = \left[n_{1_{GRU}}, n_{2_{GRU}}, \ldots, n_{l_{GRU}}, \ldots, n_{L_{GRU}}\right]$
$\mathbf{A}_{GRU}$	Vecteur contenant les fonctions d'activation dans chaque couche cachée, $\mathbf{A}_{GRU} = \left[a_{1_{GRU}}, a_{2_{GRU}}, \ldots, a_{l_{GRU}}, \ldots, a_{L_{GRU}}\right]$
$\mathbf{A}^{rec}_{GRU}$	Vecteur contenant les fonctions d'activation récurrentes dans chaque couche cachée, $\mathbf{A}_{GRU} = \left[a_{1^{rec}_{GRU}}, a_{2^{rec}_{GRU}}, \ldots, a_{l^{rec}_{GRU}}, \ldots, a_{L^{rec}_{GRU}}\right]$
$\mathbf{R}_{GRU}$	Vecteur contenant les taux de *dropout* dans chaque couche cachée, $\mathbf{R}_{GRU} = \left[r_{1_{GRU}}, r_{2_{GRU}}, \ldots, r_{l_{GRU}}, \ldots, r_{L_{GRU}}\right]$
$\mathbf{R}^{rec}_{GRU}$	Vecteur contenant les taux de *recurrent dropout* dans chaque couche cachée, $\mathbf{R}_{GRU} = \left[r_{1^{rec}_{GRU}}, r_{2^{rec}_{GRU}}, \ldots, r_{l^{rec}_{GRU}}, \ldots, r_{L^{rec}_{GRU}}\right]$

1D-CNN :

Appliquer un réseau de neurones 1D-CNN pour la prédiction de série temporelle consiste à apprendre des filtres aptes à représenter les patrons récurrents dans les données et les utiliser pour prédire les valeurs futures. Différentes architectures ont été évaluées dans le cadre de ce projet. Celles qui donnent les meilleures performances seront illustrées dans le chapitre suivant. Autres que les hyperparamètres partagés avec le LSTM et GRU, le 1D-CNN possède d'autres qui lui sont spécifiques. ils sont présentés dans le tableau 3.6.

Tableau 3.6 Hyperparamètres du modèle 1D-CNN

Symbole	**Hyperparamètre**
$\mathbf{L}_{\text{Conv}}$	Nombre de couches de convolution
$\mathbf{F}$	Vecteur contenant le nombre de filtres dans chaque couche convolutionnel, $\mathbf{F} = [f_1, f_2, \ldots, f_l, \ldots, f_{L_{\text{Conv}}}]$
$\mathbf{D}_{\text{Conv}}$	Vecteur contenant la dimension de chaque filtre dans chaque couche convolutionnel, $\mathbf{D}_{\text{Conv}} = \left[d_{1_{\text{Conv}}}, d_{2_{\text{Conv}}}, \ldots, d_{l_{\text{Conv}}}, \ldots, d_{L_{\text{Conv}}}\right]$
$\mathbf{I}_{\text{Conv}}$	Vecteur contenant les incréments des filtres dans chaque couche convolutionnel, $\mathbf{I}_{\text{Conv}} = \left[i_{1_{\text{Conv}}}, i_{2_{\text{Conv}}}, \ldots, i_{l_{\text{Conv}}}, \ldots, i_{L_{\text{Conv}}}\right]$
$\mathbf{A}_{\text{Conv}}$	Vecteur contenant les fonctions d'activation dans chaque couche convolutionnel, $\mathbf{A}_{\text{Conv}} = \left[a_{1_{\text{Conv}}}, a_{2_{\text{Conv}}}, \ldots, a_{l_{\text{Conv}}}, \ldots, a_{L_{\text{Conv}}}\right]$
$\mathbf{D}_{\text{Pool}}$	Vecteur contenant la dimension de la fonction pooling dans chaque couche de Pooling, $\mathbf{D}_{\text{Pool}} = \left[d_{1_{\text{Pool}}}, d_{2_{\text{Pool}}}, \ldots, d_{l_{\text{Pool}}}, \ldots, d_{L_{\text{Pool}}}\right]$
$\mathbf{I}_{\text{Pool}}$	Vecteur contenant les incréments de la fonction pooling dans chaque couche de Pooling, $\mathbf{I}_{\text{Pool}} = \left[i_{1_{\text{Pool}}}, i_{2_{\text{Pool}}}, \ldots, i_{l_{\text{Pool}}}, \ldots, i_{L_{\text{Pool}}}\right]$
L_{FC}	Nombre de couches *Fully Connected* (FC)
$\mathbf{A}_{\text{FC}}$	Vecteur contenant les fonctions d'activation dans chaque couche FC, $\mathbf{A}_{\text{Conv}} = \left[a_{1_{\text{FC}}}, a_{2_{\text{FC}}}, \ldots, a_{l_{\text{FC}}}, \ldots, a_{L_{\text{FC}}}\right]$
$\mathbf{R}_{\text{FC}}$	Vecteur contenant le taux du *droput* dans chaque couche FC, $\mathbf{A}_{\text{Conv}} = \left[a_{1_{\text{FC}}}, a_{2_{\text{FC}}}, \ldots, a_{l_{\text{FC}}}, \ldots, a_{L_{\text{FC}}}\right]$

3.7 Entraînement et hyperparamètres

3.7.1 Séparation des données en entraînement, validation et test

Dans ce projet, le découpage de la base de données se fait comme suit : d'abord la base de données est divisée en un ensemble de données d'entraînement et un ensemble de test avec des proportions respectivement égales à 70% et 30%. Afin d'optimiser les hyperparamètres, l'ensemble de 70% est à son tour découpé en un sous-ensemble d'entraînement (80%) et un autre de validation (20%). Le modèle de prédiction est entrainé sur les données d'entraînement et l'erreur sera calculée sur l'ensemble de validation. Les hyperparamètres choisis sont ceux qui minimisent la fonction de coût sur l'ensemble de validation. Une fois déterminé, le modèle est réentraîné avec les hyperparamètres optimaux et les métriques de performance seront mesurées sur l'ensemble de test. Intuitivement, l'ensemble de validation détermine dans quelle mesure le modèle produit de bonnes prédictions pour de nouvelles données. Ce genre de découpage, illustré dans la figure 3.13, est largement recommandé dans la littérature. En effet, il favorise la généralisation. Cette

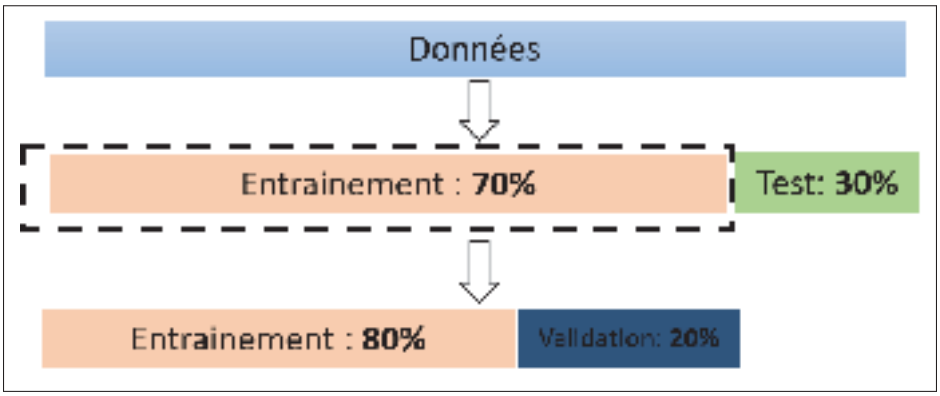


Figure 3.13 Découpage de la base de données

dernière concerne la capacité du modèle à produire des prédictions robustes sur de nouvelles données. En d'autres termes les chances de surapprentissage seront réduites considérablement avec cette séparation de données. Rappelons qu'un modèle est dit en surapprentissage s'il a de bonnes performances durant l'entraînement, mais qu'elles se dégradent énormément sur l'ensemble de validation. Le surapprentissage est causé principalement par la complexité

exagérée du modèle. Il y a un compromis entre la minimisation de l'erreur de prédiction et l'augmentation de la complexité du modèle. Ceci peut être traduit par les deux courbes présentées dans la figure 3.14. L'erreur de prédiction sur les deux ensembles d'entraînement et de validation diminue ensemble jusqu'à un montent où l'erreur sur la validation commence à augmenter. À cet instant précis, le nombre d'époques est optimal et il est préférable d'arrêter l'entraînement pour garantir une bonne généralisation du modèle. La formation de l'ensemble d'entraînement se fait comme illustré dans la figure 3.15. La même méthode s'applique pour former les ensembles de validation et de test. Supposons qu'il y'en a un vecteur contenant un nombre $N_{entrainement}$ d'échantillons en totale. Il sera découpé en plusieurs sous-ensembles de vecteurs dont la taille de chacun est égale à $N_{entree} + N_{sortie}$. Ces derniers représentent respectivement la taille de la fenêtre historique donnée en entrée au modèle et le nombre d'échantillons prédits. Chaque fois, le découpage avance d'un seul échantillon pour former le vecteur suivant jusqu'à atteindre le dernier point. L'ensemble de ces vecteurs constituent la base de données d'entraînement.

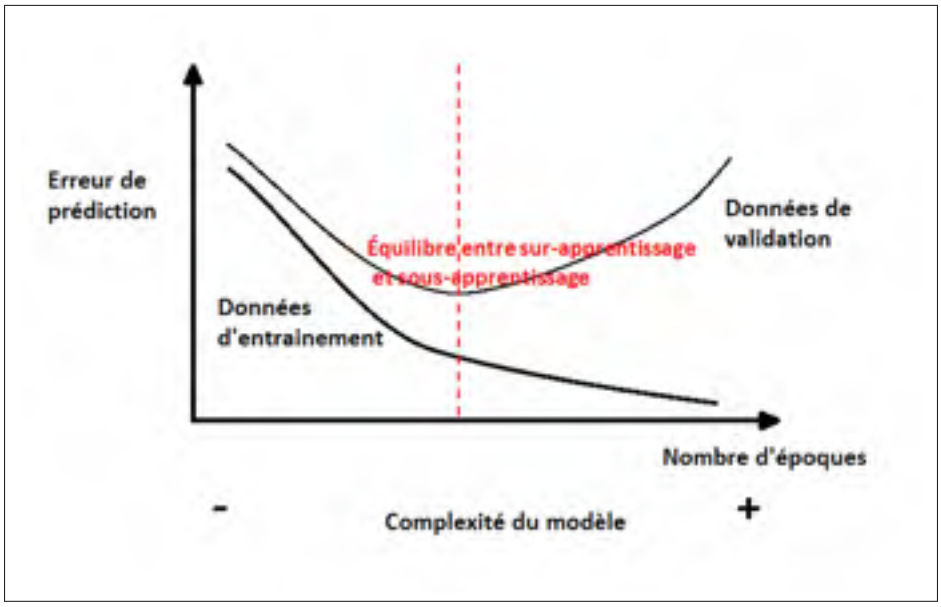


Figure 3.14 Compromis entre le surapprentissage et le sous-apprentissage

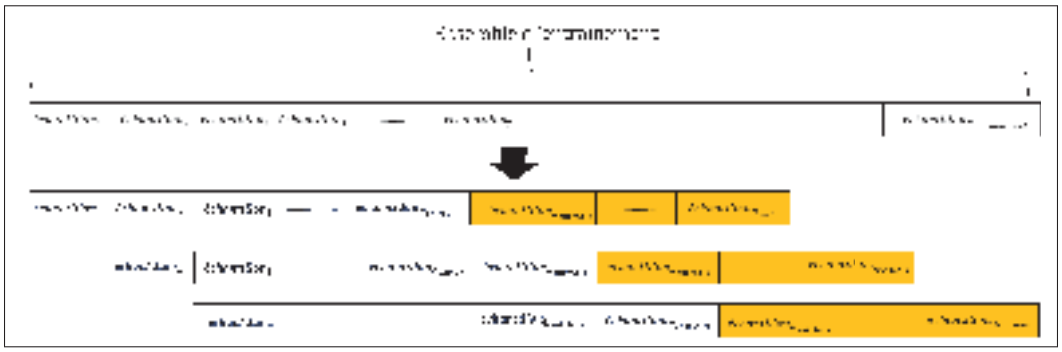


Figure 3.15 Processus de découpage de l'ensemble d'entraînement

3.7.2 Stratégie d'entraînement

Durant l'entraînement et à chaque instant t, le réseau de neurones reçoit un ensemble N_{entree} d'échantillons et prédit les N_{sortie} prochains points. Le premier ensemble est appelé fenêtre historique ou aussi fenêtre d'observation. Elle permet au réseau de neurones de capturer les changements locaux dans les données les plus récentes. La taille de cette fenêtre est un hyperparamètre important pour garantir une bonne qualité de prédiction. En effet, un modèle entrainé sur des fenêtres historiques coulissantes contenant un large nombre d'observations peut mener à des prédictions dont les tendances sont similaires à la tendance générale de l'ensemble d'entraînement. Dans ce cas, de grands écarts entre les valeurs réelles et celles prédites peuvent apparaitre. Au contraire, une fenêtre d'observation de taille très petite ne peut pas refléter correctement la tendance générale, mais capture plutôt le bruit. Dans ce stade du projet, une taille fixe a été privilégiée pour chaque circuit étudié. Cependant, la meilleure façon est de procéder avec une taille de fenêtre dynamiquement adaptée. Ce point sera consolidé dans la partie perspective. Quant à la taille optimale, elle sera déterminée avec la technique de validation croisée.

Le nombre d'époques et le *batch size* seront détaillés dans ce qui suit. Étant donné que la descente de gradient utilisée pour optimiser l'entraînement est un processus itératif et que la quantité de données disponibles est limitée, la mise à jour des poids avec un seul passage sur l'ensemble d'entraînement s'avère généralement insuffisante. Pour cette raison, le réseau de neurones reçoit

la base de données d'entraînement en entier plusieurs fois, chaque fois correspondant à une époque. Ce dernier ensemble peut être aussi suffisamment grand pour le faire passer en une seule fois au réseau de neurones. La pratique la plus commune est de le diviser en plusieurs lots avec une taille prédéfinie (‘*batch size*’ en anglais). Le batch size et le nombre d'époques précédemment mentionné sont deux hyperparamètres importants. En effet, un grand nombre d'époques favorise le surapprentissage. De plus, choisir des lots de petite taille ajoute un aspect stochastique à l'optimisation pouvant améliorer la capacité du réseau à apprendre en dépit d'un temps d'entraînement élevé.

Durant l'entraînement, deux techniques de régularisation dites ‘*dropout*’ (Srivastava, Hinton, Krizhevsky, Sutskever & Salakhutdinov, 2014) et *‘recurrent dropout‘* (Semeniuta, Severyn & Barth, 2016) sont utilisées pour réduire le surapprentissage. La première permet de retirer stochastiquement des unités ainsi que leurs connexions entrantes et sortantes des couches cachées. Chaque unité est conservée avec une probabilité p (appelé aussi taux du *dropout*) dont la valeur est optimisée à l'aide d'une validation croisée ou fixée à 0.5 comme recommandé dans la littérature. Ceci dit, le *dropout* ne peut pas être appliqué aux couches d'entrée et de sortie du réseau neuronal pour ne pas perdre des informations utiles. Cette technique, non spécifique à un domaine d'application peut améliorer les performances de réseaux de neurones, mais également augmenter le temps d'entraînement. Quant au technique de *recurrent dropout*, il permet de retirer stochastiquement des neurones directement dans les connexions récurrentes. Dans ce cas, la mémoire à long termes est mieux conserver par rapport au *dropout*. Les travaux de (Semeniuta *et al.*, 2016) démontrent l'efficacité d'utiliser le *dropout* couplé avec le *recurrent dropout* pour un réseau de neurones LSTM.

3.7.3 Fonction de coût

L'entraînement d'un réseau de neurones vise à minimiser la fonction de coût et au meilleur de cas faire converger le problème vers le minimum global. Les deux fonctions qui ont été considérées dans ce projet sont le RMSE et MAE. En optimisant le RMSE, le modèle essayera de fournir des prédictions en moyenne correctes. Minimiser le MAE conduira à avoir des prédictions avec

autant de chance de surestimer que de sous-estimer les valeurs réelles. Il n'y a pas de réponse définitive par rapport au type de la métrique à optimiser. Si la base de données contient de nombreuses valeurs aberrantes, il est recommandé d'utiliser le MAE comme fonction de coût. Si l'utilisation du MAE entraine un biais élevé, il est préférable d'optimiser le RMSE.

3.7.4 Optimisation de l'entraînement

L'optimisation de l'entraînement se fait avec l'algorithme de descente de gradient ou une de ses variantes. Les plus communes en pratique sont RMSProp et Adam. Ces derniers seront tous évalués dans le cadre de ce projet.

3.7.4.1 RMSProp

RMSProp un algorithme au taux d'apprentissage adaptatif (Schaul, Antonoglou, & Silver, 2013). Un ajustement automatique du taux d'apprentissage se fait à chaque itération. L'avantage principal du RMSProp est qu'il ne cumule pas les gradients issus de toutes les itérations précédentes pendant la mise à jour des poids. Faisant cela, le taux d'apprentissage continue de baisser à chaque itération jusqu'au devenir infinitisimalement petit au point de ne plus apporter de nouvelles connaissances aux poids. Plutôt, RMSProp restreint la fenêtre des gradients accumulés à une taille bien définie ne renfermant que les valeurs les plus récentes. Une moyenne mobile exponentielle des carrés des gradients précédents est ainsi appliquée au lieu de stocker les w anciennes valeurs. La moyenne courante à l'instant t représentée par $\mathrm{E}\left[g^2\right]_t$ dans l'équation suivante dépend de la moyenne à l'instant $t-1$ et du gradient à l'instant t. Il est recommandé que le paramètre δ soit égal à 0.9. Le taux d'apprentissage initial est représenté par η. Sa valeur par défaut est égale à 0.001.

$$\mathrm{E}\left[g^2\right]_t = \delta\,\mathrm{E}\left[g^2\right]_{t-1} + (1-\delta)g_t^2 \tag{3.10}$$

$$w_{t+1} = w_t - \frac{\eta}{\sqrt{\mathrm{E}\left[g^2\right]_t + \epsilon}} g_t \tag{3.11}$$

Diviser le gradient de la fonction coût par la racine carrée de la moyenne quadratique glissante accélère la convergence.

3.7.4.2 Adam

Adam (*Adaptive Moment Estimation*) est l'une des variantes de la descente de gradient les plus récentes et les plus efficaces (Kingma & Ba, 2014) dans les applications du monde réell. Tout comme RMSProp, Adam détermine un taux d'apprentissage adaptatif pour chaque paramètre w_i en estimant le moment d'ordre 1 (traduit par la moyenne) et le moment d'ordre 2 (traduit par la variance) des gradients précédents. La moyenne notée m_t n'est autre que la somme de décomposition moyenne des gradients précédents, comme exprimé dans l'équation ci-dessous. Quant à la variance, elle est égale à la somme de décomposition moyenne des gradients précédents élevés au carré. Elle est notée par v_t .

$$m_t = \gamma_1 m_{t-1} + (1 - \gamma_1) g_t \tag{3.12}$$

$$v_t = \gamma_2 v_{t-1} + (1 - \gamma_2) g_t^2 \tag{3.13}$$

Les paramètres γ_1 et γ_2 sont des coefficients de décroissances dont les valeurs sont généralement proches de 1. Les valeurs proposées dans la littérature pour γ_1 et γ_2 sont respectivement 0.9 et 0.999. La moyenne m_t et la variance v_t sont initialisés comme des vecteurs de 0. Durant les premières itérations, ces derniers vecteurs sont biaisés à zéro. Ainsi, une correction de l'estimation de la moyenne et variance a été proposée. Elles sont exprimées par :

$$\hat{m}_t = \frac{m_t}{1 - \gamma_1^t} \tag{3.14}$$

$$\hat{v}_t = \frac{v_t}{1 - \gamma_2^t} \tag{3.15}$$

La mise à jour des poids dans l'algorithme d'Adam se fait comme suit :

$$w_{t+1} = w_t - \frac{\eta}{\sqrt{\hat{v}_t + \epsilon}} \hat{m}_t \tag{3.16}$$

Le paramètre η est le taux d'apprentissage initial. L'utilisation de la valeur 0.001 est très fréquente dans la littérature. Cependant, elle reste un hyperparamètre à optimiser.

3.7.5 Validation croisée pour les données temporelles

La validation croisée est une technique largement appliquée pour optimiser les hyperparamètres et justifier la robustesse du modèle de prédiction : les métriques de performances sont moyennées sur plusieurs ensembles de tests. En effet, l'évaluation de la qualité de prédiction sur un seul ensemble de test peut résulter en des performances biaisées. Cet ensemble arbitrairement choisi peut être plus facile (ou difficile) à prédire que la distribution générale des données, donnant une évaluation trop optimiste (ou pessimiste). Un type bien particulier de validation croisée doit être appliqué aux données temporelles. Elle doit entre autres préserver l'ordre chronologique dans les données. La procédure de cette validation croisée est illustrée dans la figure 3.16. La base de données est découpée en 3 parties ($k = 3$) sur lesquels l'entraînement, le choix des hyperparamètres optimaux et le calcul des métriques de performance (à savoir RMSE) sur l'ensemble de test sont appliqués à chaque fois. Une moyenne globale des métriques de performance ainsi que leur écart type seront fournies en dernière étape. Le découpage se fait de façon à ce que, pour $k = 3$, la totalité de la base de données est utilisée pour l'entraînement, validation et test. Le choix de ce nombre de partitions, basé sur la taille des données disponibles, est recommandé dans la littérature.

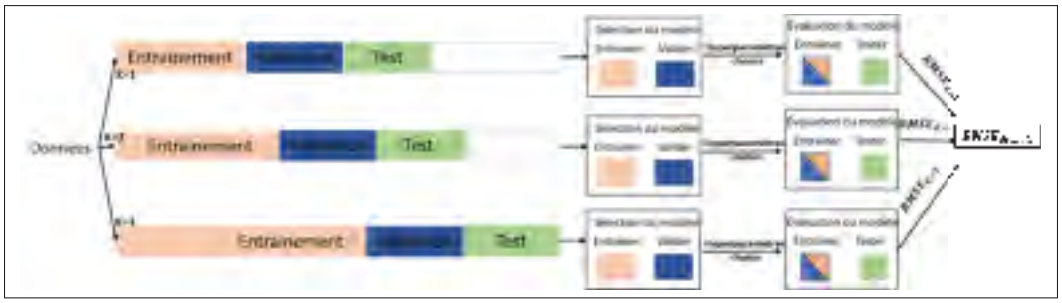


Figure 3.16 Validation croisée pour les données temporelles ($K = 3$)

3.7.6 Optimisation des hyperparamètres

La détermination des hyperparamètres optimaux peut se faire à l'aide d'une procédure de grille de recherche (en anglais '*grid search*') ou tout simplement par une recherche aléatoire. La première méthode consiste à choisir d'avance des plages de valeurs à parcourir pour chacun des hyperparamètres. Toutes les combinaisons possibles d'hyperparamètres sont ensuite construites par l'algorithme dans une sorte de grille. La combinaison d'hyperparamètres qui donne les meilleures performances sur l'ensemble de validation est sélectionnée. Ceci se dit, le nombre de combinaisons candidates dépend du nombre d'hyperparamètres dans le modèle et de la plage des valeurs suggérée. De ce fait, cette procédure requiert une durée d'exécution assez élevée.

Dans la recherche aléatoire, une distribution de valeurs admissibles est attribuée à chaque hyperparamètre dans le modèle. Plusieurs entraînements sont ensuite effectués tout en échantillonnant aléatoirement les valeurs des hyperparamètres parmi les valeurs de la distribution. Les hyperparamètres ayant obtenu la meilleure erreur de généralisation seront sélectionnés. Selon l'étude de (Bergstra & Bengio, 2012), la recherche aléatoire est beaucoup plus efficace comparé à la recherche en grille. Cette deuxième méthode sera appliquée dans le cadre de ce projet.

3.7.7 Apprentissage par transfert pour la prédiction de la qualité de performance

Dans cette section, la stratégie d'entraînement des modèles de réseaux de neurones en utilisant l'apprentissage par transfert est expliquée. La technique utilisée est inspirée de l'article (He *et al.*, 2019). Considérons trois circuits notés X, Y et Z tel que le premier est le circuit cible et les deux derniers sont les sources. Le transfert de connaissance se fait à partir des deux bases de données sources notées D_s^y et D_s^z (correspondant respectivement aux circuits Y et Z) envers le circuit X (dont les données sont notées D_c^x). Supposons aussi que D_s^y est plus similaire à la base de données cibles D_c^x que D_s^z en utilisant la mesure de similarité DTW. Comme illustré dans la figure 3.17, la base de données D_s^y est utilisée pour préentraîner le réseau de neurones. Ensuite, les poids de la première couche sont gelés et le modèle est entraîné en utilisant la base de données D_s^z. Enfin, l'ensemble d'entraînement de la base de données cibles D_c^x sert

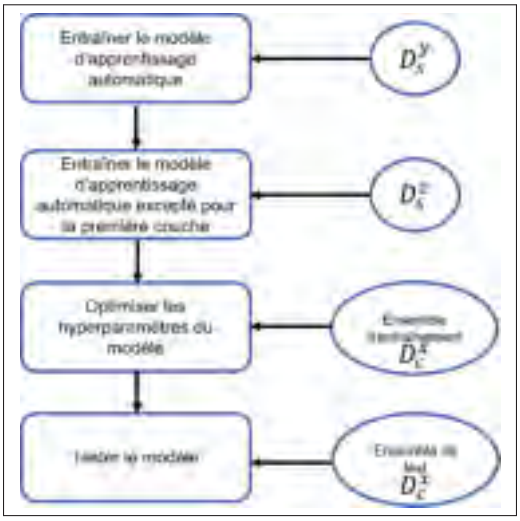


Figure 3.17 Stratégie d'entraînement pour l'apprentissage par transfert avec deux sources
Adaptée de He *et al.* (2019)

à déterminer les hyperparamètres optimaux. En tout, la première couche du modèle permet d'apprendre les patrons du circuit Y. Quant à la deuxième couche, elle permet l'apprentissage des patrons des deux circuits Y et Z. Notons que le nombre de couches cachées utilisé dans le modèle doit être impérativement supérieur ou égal au nombre de sources.

Les circuits Y et Z sont les plus similaires au circuit cible. En d'autres termes, ils ont les mesures de similarité DTW les plus petites parmi l'ensemble des 150 circuits étudiés dans le cadre de ce projet. De ce fait, la valeur moyenne des deux distances DTW correspondant aux circuits sources avec le circuit cible est utilisée comme critère principal pour les identifier. Le calcul de cette moyenne est donnée par l'équation 3.17. Plus la valeur de mDTW est petite, plus similaires sont les données sources et cible.

$$mDTW = \frac{DTW(D_s^y, D_c^x) + DTW(D_s^z, D_c^x)}{2} \tag{3.17}$$

3.8 Outils

Le projet est entièrement développé en langage Python. Les réseaux de neurones sont construits avec la librairie Keras (Chollet et al., 2015), qui se base essentiellemt sur Tensorflow (Abadi, Barham, Chen, Chen, Davis, Dean, Devin, Ghemawat, Irving, Isard et al., 2016). Quant aux *baselines*, ils sont développés avec la librairie pmdarima (Smith et al., 2017). Le code source de cette dernière a été modifié pour favoriser la rapidité des algorithmes. Finalement, toutes les simulations sont effectuées sur une machine Windows10 utilisant un processeur Intel(R) Core(TM) $i7-6700$ avec une mémoire RAM de 16 GB.

CHAPITRE 4

RÉSULTATS ET DISCUSSION

4.1 Introduction

Dans ce chapitre, une analyse comparative des différentes méthodes étudiées est présentée pour les circuits A, B, C et D. Ces derniers sont évalués en utilisant les métriques de performance biais, RMSE, MAE et R^2 selon un scénario qui est décrit plus tard dans la section. Par la suite, des expériences ont été effectuées sur la méthode *stateful* LSTM, dont le choix est justifié dans la section. Ces analyses permettent d'étudier la robustesse de la méthode selon des scénarios donnés. Commençons par évaluer la capacité prédictive du modèle pour un horizon de 96 heures. Viennent ensuite l'analyse de l'effet de la prédiction multivariée et de l'effet de l'apprentissage par transfert.

4.2 Comparaison des résultats des différentes méthodes étudiées

Dans cette section, la comparaison des différentes méthodes de prédiction est effectuée, et ce en se basant sur les métriques de performances suivantes : biais, MAE, RMSE, et R^2. Rappeleons que les modèles sont testés sur les circuits A, B, C et D dont l'analyse descriptive est présentée dans la section 3.2. Dans ce qui suit, le scénario de la comparaison est expliqué et les hyperparamètres optimaux sont mentionnés pour chaque modèle étudié.

Il s'agit ici de prédire la qualité de performance des réseaux optiques opérationnels traduite par le SNR pour un horizon de prédiction de 24 heures. Le choix de cet horizon est convenu avec le partenaire industriel. La prédiction est univariée : seules les données historiques de SNR sont utilisées comme *feature* des modèles. Des données manquantes et d'autres aberrantes existent dans nos données temporelles. Les valeurs aberrantes sont supprimées selon la procédure expliquée dans la section 3.3.2. Ces dernières seront considérées comme des valeurs manquantes pour le reste de la section et reçoivent le traitement suivant : si la fréquence d'apparition des données manquantes par mois ne dépasse pas 10% de l'ensemble des données manquantes

du circuit, elles sont supprimées. Dans le cas contraire, ces valeurs sont imputées avec un réseau LSTM de type *stateful* (section 3.3.1). En termes de répartitions de données, les mêmes ensembles d'entraînement, validation et test sont utilisés pour tous les modèles évalués. Les détails concernant chacun de ces modèles sont fournis ci-dessous :

***Baseline* :**

Le modèle de référence est un ARIMA ou SARIMA dépendamment de l'existence ou non de la saisonnalité. De ce fait, ARIMA est utilisé pour les circuits B, C et D où la saisonnalité est très faible ou nulle. Par contre, c'est le modèle SARIMA qui est considéré pour le circuit A, ayant une saisonnalité de 24 heures. Étant donné que des tests de stationnarité sont déjà effectués (ADF et KPSS), l'ordre de stationnarité est connu pour tous les circuits : $d = 1$. En outre, le test de Canova-Hansen (Canova & Hansen, 1995) est utilisé dans le but de déterminer l'ordre de différenciation saisonnier. Quant au paramètre m (nombre d'observations dans chaque saison), il est connu à priori à partir des données. Finalement, pour identifier les paramètres optimaux p, q, P, Q de ces modèles, les séries temporelles sont divisées en ensemble d'entraînement et de test avec les proportions de 70% et 30% respectivement. Ensuite, l'ensemble d'entraînement est de nouveau divisé en entraînement et validation avec les proportions de 80% et 20% (figure 3.15). Les paramètres qui donnent l'erreur de prédiction la plus faible sur l'ensemble de validation sont ceux étant optimaux. Les valeurs obtenues sont ensuite validées avec celles déterminées en utilisant les corrélogrammes et les corrélogrammes partiels de tous les circuits. Rappelons que les paramètres des modèles ARIMA et SARIMA s'écrivent respectivement, ARIMA(p,d,q) et SARIMA(p,d,q)(P,D,Q,m). Les résultats sont les suivants :

- **Circuit A :** SARIMA(2,1,1)(1,1,2,96) ;
- **Circuit B :** ARIMA(1,1,2) ;
- **Circuit C :** ARIMA(1,1,2) ;
- **Circuit D :** ARIMA(4,1,1).

Suite à la détermination de ces paramètres, le modèle est réentraîné encore une fois sur les ensembles d'entraînement et validation combinés. L'évaluation des performances est effectuée

sur l'ensemble de test. En vue d'intégrer les données de tests les plus récentes, les modèles ARIMA sont mis à jour chaque heure. Cette opération n'est pas coûteuse en matières de temps et de capacité de calcul. Cependant, le modèle SARIMA requiert beaucoup plus de temps durant l'entraînement. Ce problème est bien connu en pratique surtout en ayant un paramètre m large. Pour cette raison, le modèle SARIMA n'est mis à jour que deux fois durant le test à des intervalles réguliers. C'est pour cela aussi que le modèle ARIMA avec mise à jour a été préféré pour les deux circuits B et C ayant une faible saisonnalité. Ceci a été bel et bien validé avec les simulations.

Réseaux de neurones

Dans ce scénario, les données sont différenciées en premier lieu. Ensuite, elles sont normalisées conformément à la section 3.3.3. Durant l'entraînement, l'algorithme Adam est celui utilisé pour minimiser la fonction de coût, soit la racine carrée de l'erreur quadratique moyenne. Enfin, les modèles ne sont pas mis à jour durant la phase de test. En effet, pour qu'elle soit efficace, cette opération requiert une nouvelle optimisation des hyperparamètres.

Les tableaux 4.1, 4.2, 4.3 et 4.4 détaillent respectivement les hyperaparamètres correspondant aux modèles *stateful* LSTM, *stateless* LSTM, *stateful* GRU et 1D-CNN pour les différents circuits étudiés. Les hyperparamètres choisis sont ceux minimisant l'erreur sur l'ensemble de validation. Dans ce qui suit, une comparaison des résultats obtenus est proposée pour chacun des modèles étudiés. Les figures 4.1, 4.2, 4.3 et 4.4 montrent l'évolution du biais, MAE, RMSE et R^2 en fonction de l'horizon de prédiction, respectivement pour les circuits A, B, C et D. Pour un horizon de prédiction donné, c'est la moyenne des métriques sur toutes les observations qui est illustrée dans les figures. Rappelons que plus le RMSE et MAE sont proches de 0, meilleure est la prédiction. De plus, une valeur de R^2 proche de 1 implique un modèle plus performant. Pour évaluer la robustesse des modèles, des métriques globales ont été calculées en utilisant une validation croisée avec $k = 3$ (section 3.7.5). La démarche se fait comme suit : pour chaque instant prédit, la médiane des métriques est calculée sur les trois ensembles de tests. Ayant déterminé la liste des médianes pour tous les instants prédits, la médiane globale et la variation standard sont mesurées sur ces dernières valeurs. Ces métriques sont résumées dans les tableaux

Tableau 4.1 Valeurs des hyperparamètres pour les circuits étudiés : modèle *stateful* LSTM

Symbole	**Circuit A**	**Circuit B**	**Circuit C**	**Circuit D**
L_{LSTM}	2	2	1	2
$\mathbf{N}_{LSTM}$	[100, 50]	[50, 50]	[100]	[20, 20]
$\mathbf{A}_{LSTM}$	[tanh, tanh]	[tanh, tanh]	[tanh]	[tanh, tanh]
$\mathbf{A}^{rec}_{LSTM}$	$[\sigma, \sigma]$	$[\sigma, \sigma]$	$[\sigma]$	$[\sigma, \sigma]$
$\mathbf{R}_{LSTM}$	[0.5, 0.5]	[0.5, 0.5]	[0.5]	[0.5, 0.5]
$\mathbf{R}^{rec}_{LSTM}$	[0.1, 0.2]	[0.1, 0.1]	[0.1]	[0.2, 0.2]
n_{batch}	256	8	64	64
n_{epochs}	100	70	60	75
$\alpha_{initial}$	0.0001	0.005	0.001	0.001
H	96	288	192	288

Tableau 4.2 Valeurs des hyperparamètres pour les circuits étudiés : modèle *stateless* LSTM

Symbole	**Circuit A**	**Circuit B**	**Circuit C**	**Circuit D**
L_{LSTM}	2	1	1	2
$\mathbf{N}_{LSTM}$	[50, 50]	[300]	[150]	[250, 250]
$\mathbf{A}_{LSTM}$	[tanh, tanh]	[tanh]	[tanh]	[tanh, tanh]
$\mathbf{A}^{rec}_{LSTM}$	$[\sigma, \sigma]$	$[\sigma]$	$[\sigma]$	$[\sigma, \sigma]$
$\mathbf{R}_{LSTM}$	[0.5, 0.5]	[0.5]	[0.5]	[0.5, 0.5]
$\mathbf{R}^{rec}_{LSTM}$	[0.2, 0.1]	[0.3]	[0.2]	[0.2, 0.2]
n_{batch}	512	480	512	512
n_{epochs}	80	75	50	25
$\alpha_{initial}$	0.001	0.0025	0.001	0.001
H	96	96	192	288

4.5, 4.6, 4.7, 4.8 et ce pour les circuits A, B , C et D. Chaque métrique est représentée dans les tableaux sous la forme suivante : (médiane globale, variation standard).

Circuit A :

Globalement, c'est le modèle *stateful* LSTM qui produit les meilleures résultats. Les modèles 1D-CNN, *stateless* LSTM et *stateful* GRU ont des résultats aussi comparables avec ce dernier. Cependant, le modèle SARIMA a des performances médiocres, avec une faible variation en

Tableau 4.3 Valeurs des hyperparamètres pour les circuits étudiés : modèle *stateful* GRU

Symbole	**Circuit A**	**Circuit B**	**Circuit C**	**Circuit D**
L_{GRU}	1	2	1	2
$\mathbf{N}_{GRU}$	[100]	[75, 50]	[150]	[250, 250]
$\mathbf{A}_{GRU}$	[tanh]	[tanh, tanh]	[tanh]	[tanh, tanh]
$\mathbf{A}^{rec}_{GRU}$	[σ]	[σ, σ]	[σ]	[σ, σ]
$\mathbf{R}_{GRU}$	[0.5]	[0.5, 0.5]	[0.5]	[0.5, 0.5]
$\mathbf{R}^{rec}_{GRU}$	[0.25]	[0.1, 0.2]	[0.2]	[0.2, 0.2]
n_{batch}	64	32	64	64
n_{epochs}	80	100	70	75
$\alpha_{initial}$	0.001	0.001	0.001	0.0001
H	96	96	192	480

Tableau 4.4 Valeurs des hyperparamètres pour les circuits étudiés : modèle 1D-CNN

Symbole	**Circuit A**	**Circuit B**	**Circuit C**	**Circuit D**
$\mathbf{L}_{Conv}$	2	2	2	1
$\mathbf{F}$	[64, 32]	[32, 16]	[64, 32]	[16]
$\mathbf{D}_{Conv}$	[20, 10]	[15, 10]	[15, 15]	[10]
$\mathbf{I}_{Conv}$	[8, 4]	[4, 4]	[4, 4]	[4]
$\mathbf{A}_{Conv}$	[relu, relu]	[relu, relu]	[relu, relu]	[relu]
$\mathbf{D}_{Pool}$	[2, 2]	[2, 2]	[4, 4]	[2]
$\mathbf{I}_{Pool}$	[2, 2]	[2, 1]	[1, 1]	[1]
L_{FC}	2	2	1	2
$\mathbf{A}_{FC}$	[relu]	[relu]	[tanh]	[relu]
$\mathbf{R}_{FC}$	[0.5, 0]	[0.5, 0]	[0.5, 0]	[0]
n_{batch}	32	64	32	8
n_{epochs}	150	200	50	200
$\alpha_{initial}$	0.001	0.0001	0.001	0.00008
H	96	96	192	96

fonction de l'horizon de prédiction. En effet, SARIMA n'est mis à jour que 2 fois durant la phase de test. Les prédictions estimées sont très proches l'une des autres. Ayant des valeurs de R^2 négatives pour tout l'horizon de prédiction, il est certain que la relation entre l'entrée du modèle et la variable prédite n'est pas linéaire. La direction de la moyenne de l'erreur varie en

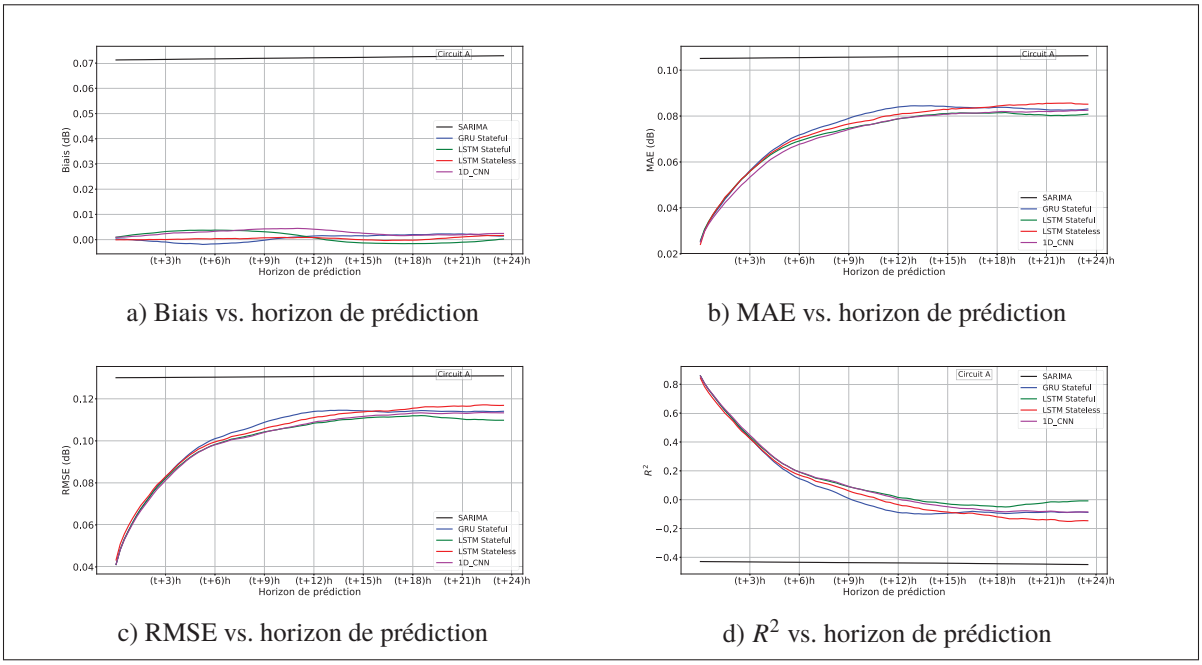
fonction de l'horizon de prédiction pour tous les réseaux de neurones. Les courbes de RMSE et MAE croissent en fonction de l'horizon de prédiction jusqu'à $t + 15$ heures. Ils se stabilisent ou presque, au-delà de cet horizon. Les valeurs de RMSE obtenues sont strictement supérieures à ceux du MAE. Ainsi, les erreurs de prédiction n'ont pas toute la même amplitude pour différents horizons de prédiction. Quant aux valeurs de R^2, elles sont positives jusqu'à $t + 9$, $t + 11$ et $t + 12$ respectivement pour *stateful* GRU, LSTM *stateless*, et les deux modèles 1D-CNN et *stateful* LSTM. Après ces horizons, les réseaux de neurones sont moins performants qu'une simple agrégation des données avec la moyenne. Seule la prédiction à court terme est possible pour le circuit A. Suite à l'application d'une validation croisée (avec $k = 3$), les performances globales et les déviations standard des métriques MAE, RMSE et R^2 ont été calculé. Comme le montre le tableau 4.5, les trois ensembles de tests ont des performances similaires.

Tableau 4.5 Évaluation de la robustesse des modèles étudiés : circuit A

Modèle	**MAE (dB)**	**RMSE (dB)**	$\boldsymbol{R^2}$
***Stateful* LSTM**	$(0.077, 0.000758)$	$(0.1066, 0.0011)$	$(0.0248, 0.000898)$
***Stateful* GRU**	$(0.8321, 0.00064)$	$(0.11489, 0.00487)$	$(-0.0821, 0.0017)$
***Stateless* LSTM**	$(0.0805, 0.00049)$	$(0.11, 0.0004)$	$(-0.0283, 0.00359)$
1D-CNN	$(0.07825, 0.00015)$	$(0.1095, 0.00074)$	$(0.01351, 0.00062)$
SARIMA	$(0.1121, 0.0082)$	$(0.1362, 0.0047)$	$(-0.437, 0.0012)$

Circuit B :

Les résultats obtenus pour le circuit B montrent que le modèle ARIMA surpasse tous les autres modèles de prédiction évalués en termes de MAE, RMSE et R^2. Viennent ensuite les modèles *stateful* LSTM, *stateful* GRU, *stateless* LSTM et enfin 1D-CNN. Sauf pour le 1D-CNN, les modèles ont tendance à surestimer la qualité de performance (biais postif). Ce n'est pas tout à fait le cas pour le 1D-CNN qui présente un biais négatif au-delà de $t + 6$ heures. En ce qui concerne les valeurs de MAE et RMSE, elles augmentent en fonction de l'horizon de prédiction jusqu'à 12 heures comme c'est illustré dans la figure 4.2. Juste après, une légère baisse est notée entre $t + 12$ et $t + 21$ heures. Un retour à l'augmentation est observée en dernier lieu. À $t + 24$ heures, le modèle ARIMA atteint une amélioration maximale de 0.0015 dB et de 0.0038 dB



a) Biais vs. horizon de prédiction

b) MAE vs. horizon de prédiction

c) RMSE vs. horizon de prédiction

d) R^2 vs. horizon de prédiction

Figure 4.1 Comparaison des méthodes étudiées pour le circuit A

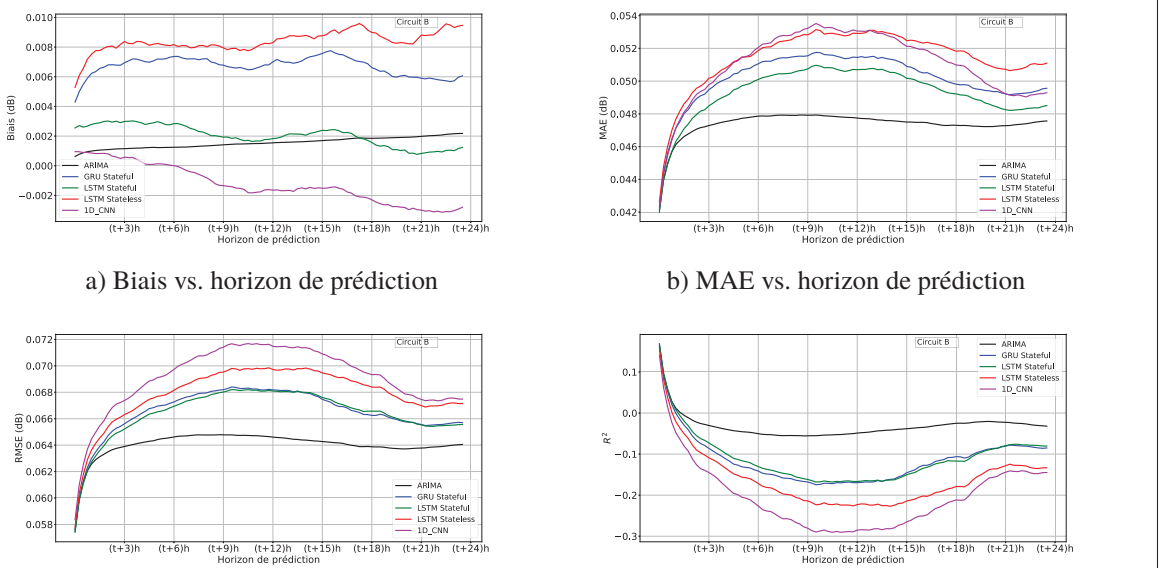
respectivement en termes de MAE et RMSE, et ce comparé au modèle 1D-CNN. La même allure de courbe est obtenue avec la métrique R^2, mais avec une tendance baissière. Selon les valeurs de R^2 obtenues, les performances de tous les modèles se dégradent après 1 heure et 45 minutes. Ceci est traduit par des valeurs de R^2 qui sont négatives au-delà de cet horizon. Ainsi une simple moyenne des données produit une meilleure prédiction que tous les modèles étudiés. Durant l'entraînement des différents réseaux de neurones, les erreurs de prédiction enregistrées sont élevées. Cependant, ce n'est pas souhaitable de réduire cette erreur provenant déjà du bruit car cela peut entraîner le surapprentissage. Notons aussi que la mise à jour du modèle ARIMA est très bénéfique dans ce contexte. Elle a pu améliorer les performances du pire au meilleur modèle pour ce circuit. Finalement, des performances comparables ont été retrouvées sur tous les trois ensembles de test (tableau 4.6).

Circuit C :

Quant au circuit C, les modèles *stateful* LSTM et *stateful* GRU fournissent les meilleures performances. Viennent subséquemment les modèles *stateless* LSTM et 1D-CNN. Le dernier est

Tableau 4.6 Évaluation de la robustesse des modèles étudiés : circuit B

Modèle	MAE (dB)	RMSE (dB)	R^2
***Stateful* LSTM**	$(0.0485, 0.00038)$	$(0.0662, 0.00016)$	$(-0.12, 0.00193)$
***Stateful* GRU**	$(0.05142, 3.26e^{-5})$	$(0.0666, 0.000285)$	$(-0.127, 000329)$
***Stateless* LSTM**	$(0.05208, 1.24e^{-5})$	$(0.0687, 0.0004)$	$(-0.179, 0.0041)$
1D-CNN	$(0.05195, 2.94e^{-5})$	$(0.0701, 0.00053)$	$(-0.257, 0.0293)$
ARIMA	$(0.0472, 0.00028)$	$(0.0644, 0.00026)$	$(-0.036, 0.00249)$



a) Biais vs. horizon de prédiction

b) MAE vs. horizon de prédiction

c) RMSE vs. horizon de prédiction

d) R^2 vs. horizon de prédiction

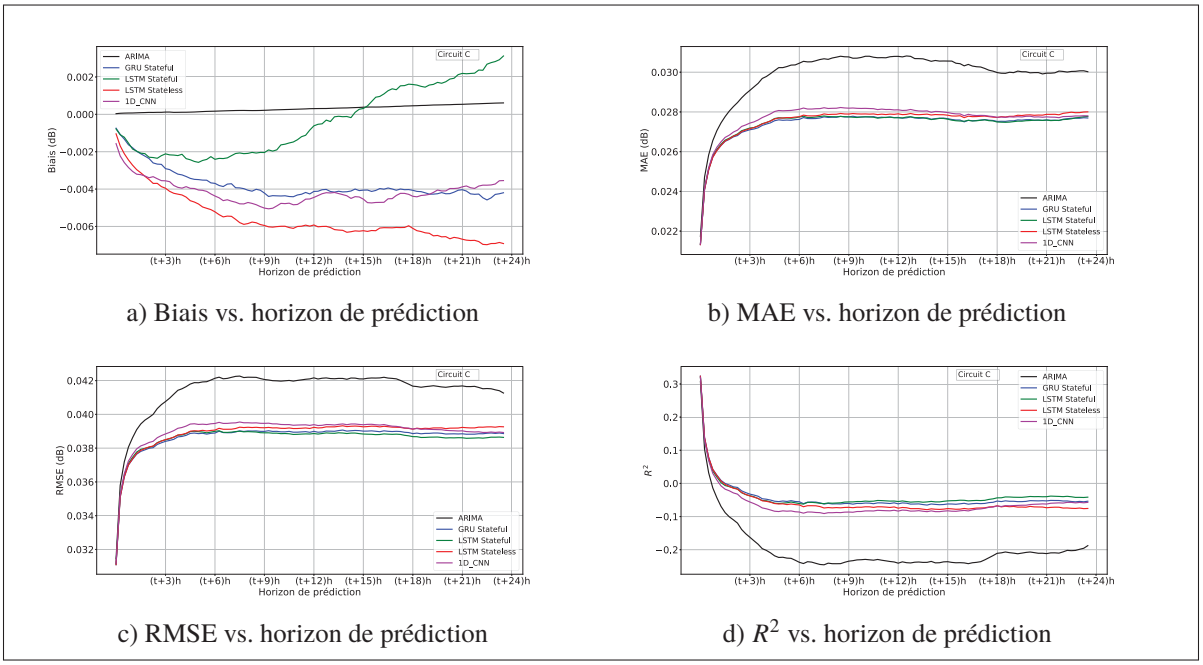
Figure 4.2 Comparaison des méthodes étudiées pour le circuit B

le modèle ARIMA. Une amélioration de 0.0022 dB, 0.0028 dB et 0.11 respectivement en termes de MAE, RMSE et R^2 est observé entre les modèles *stateful* LSTM et ARIMA. Les modèles *stateful* GRU, *stateless* LSTM et 1D-CNN ont tendance à sous-estimer la qualité de performance. En ce qui concerne *stateful* LSTM, le biais varie entre des valeurs négatives jusqu'à 13 heures et 45 minutes et d'autres positives au-delà de cet horizon. Le modèle ARIMA surestime la qualité de performance pour tout l'horizon de prédiction. Étant donnée la direction négative de la moyenne d'erreur, le modèle *stateful* GRU peut être préférable que le *stateful* LSTM, surtout qu'ils ont des performances comparables pour le reste des métriques. Tout comme le circuit

B, la prédiction s'avère difficile après 45 minutes pour tous les réseaux de neurones et après 1 heure pour le modèle ARIMA. En effet, les valeurs de R^2 descendent au dessous de zéro. Le tableau 4.7 révèle des performances comparables avec la technique de validation croisée.

Tableau 4.7 Évaluation de la robustesse des modèles étudiés : circuit C

Modèle	MAE (dB)	RMSE (dB)	R^2
Stateful LSTM	$(0.02751, 8.57e^{-5})$	$(0.038818, 6.05e^{-5})$	$(-0.0497, 0.0012)$
Stateful GRU	$(0.02768, 3.85e^{-5})$	$(0.03893, 4.29e^{-5})$	$(-0.0549, 0.0009)$
***Stateless* LSTM**	$(0.02783, 7.40e^{-5})$	$(0.03919, 0.00013)$	$(-0.0752, 0.0033)$
1D-CNN	$(0.02799, 0.00013)$	$(0.0395, 0.00011)$	$(-0.0804, 0.00377)$
ARIMA	$(0.031, 0.0016)$	$(0.0492, 0.00429)$	$(-0.231, 0.01)$



a) Biais vs. horizon de prédiction

b) MAE vs. horizon de prédiction

c) RMSE vs. horizon de prédiction

d) R^2 vs. horizon de prédiction

Figure 4.3 Comparaison des méthodes étudiées pour le circuit C

Circuit D :

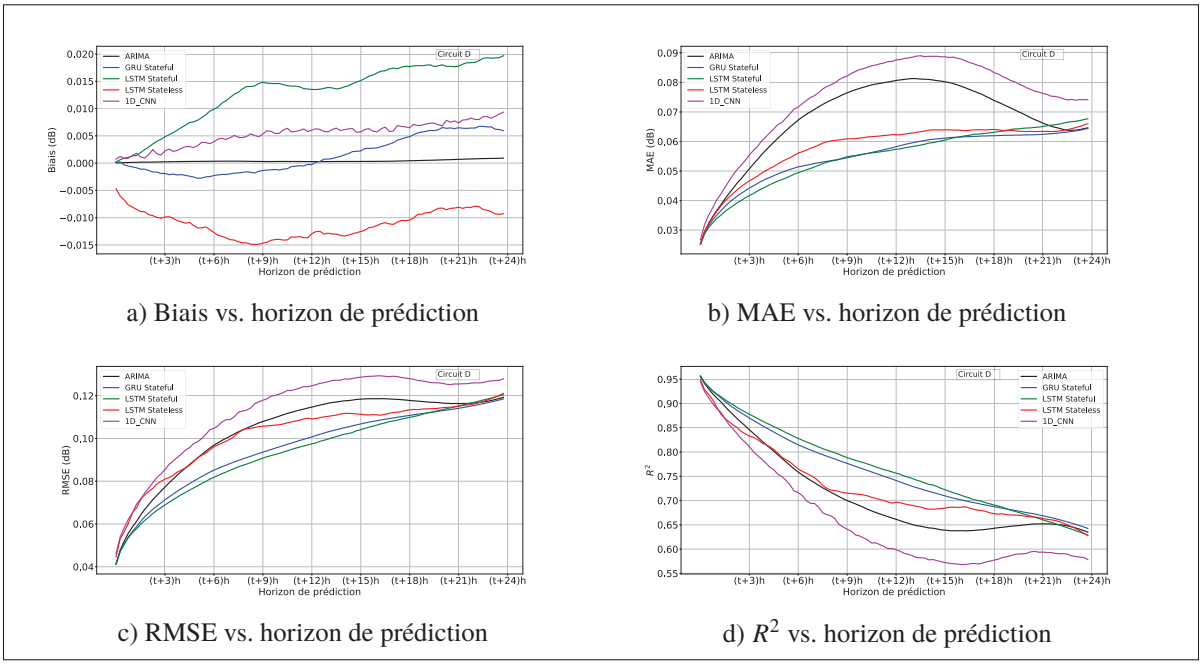
La capacité des modèles étudiés à prédire la variable SNR du circuit D est supérieure comparée aux autres circuits. En matière de R^2, le pouvoir prédictif des modèles varie selon l'horizon de prédiction et peut être classé comme suivant. Entre t et $t+2$ heures, le *stateful* LSTM, *stateful*

GRU, ARIMA et les deux modèles à la fois *stateless* LSTM et 1D-CNN donnent respectivement les meilleures valeurs. Au-delà de $t + 2$ heures, le modèle 1D-CNN est classé le dernier. À $t + 5$ heures, le modèle *stateless* LSTM surpasse l'ARIMA. Les restent respectent la même ordre qu'auparavant. Entre $t + 5$ heures et $t + 19$ heures, le classement suit cet ordre : *stateful* LSTM, *stateful* GRU, *stateless* LSTM, ARIMA et enfin 1D-CNN. À $t + 24$ heures, le *stateful* GRU produit les prédictions les plus précises. Quant à la direction de la moyenne d'erreur, le modèle ARIMA produits des biais très proches de la valeur nulle. Le *stateful* LSTM et 1D-CNN surestiment la qualité de performance. Cependant, le *stateless* LSTM la soustime. Et finalement, les biais du *stateful* GRU varient entre des valeurs négatives avant $t + 12$ heures et d'autres positives après cet horizon. En se basant sur les métriques globales calculées dans le tableau 4.8, le *stateful* LSTM fournit les meilleures prédictions, ensuite le *stateful* GRU, *stateless* LSTM ARIMA et 1D-CNN. Les valeurs de R^2 obtenues sont toutes positives indépendamment du modèle et de l'horizon de prédiction. Les prédictions estimées produisent moins d'erreurs que la valeur moyenne des données. De plus, le fait que le modèle ARIMA avec toute sa simplicité réussit à produire des résultats compétitifs révèle l'existence d'une certaine linéarité dans les données. Pour ce circuit également, les valeurs de RMSE sont supérieures aux valeurs de MAE. Ainsi, l'amplitude des erreurs n'est pas partout la même. Enfin, les modèles étudiés sont robustes selon les résultats de la validation croisée (tableau 4.8).

Tableau 4.8 Évaluation de la robustesse des modèles étudiés : circuit D

Modèle	**MAE (dB)**	**RMSE (dB)**	$\boldsymbol{R^2}$
***Stateful* LSTM**	(0.05617, 0.00157)	(0.092, 0.0031)	(0.772, 0.0089)
***Stateful* GRU**	(0.0578, 0.00167)	(0.1001, 0.001)	(0.0752, 0.010)
***Stateless* LSTM**	(0.0658, 0.0023)	(0.1091, 0.0010)	(0.706, 0.0058)
1D-CNN	(0.0789, 0.0047)	(0.11805, 0.0034)	(0.635, 0.009)
ARIMA	(0.0714, 0.0014)	(0.1098, 0.0024)	(0.672, 0.008)

Notons que la taille de la fenêtre historique d'observation a un impact significatif sur les performances des modèles, et ce pour tous les circuits. Cette dernière dépend non seulement des données, mais parfois du modèle utilisé. Parmi les autres expériences qui ont été menées durant



a) Biais vs. horizon de prédiction

b) MAE vs. horizon de prédiction

c) RMSE vs. horizon de prédiction

d) R^2 vs. horizon de prédiction

Figure 4.4 Comparaison des méthodes étudiées pour le circuit D

cette étude comparative est l'utilisation de RMSProp comme algorithme d'optimisation et de l'erreur absolue moyenne comme fonction de coût. Aucune d'elles n'a provoqué un changement considérable dans les performances des réseaux ou dans l'allure des courbes des métriques en fonction de l'horizon de prédiction. De ce fait, l'algorithme Adam et la racine carrée de l'erreur quadratique moyenne seront adoptés dans ce qui suit.

En tout, la prédiction est possible à court terme uniquement (12 heures) pour le circuit A et ce en utilisant les réseaux de neurones. Les performances du *stateul* LSTM en matières de MAE, RMSE et R^2 sont légèrement supérieurs aux autres réseaux de neurones. L'application du modèle SARIMA pour le circuit A est difficile à cause de la valeur élevée du paramètre m ($m = 96$). À cet égard, il n'est pas conseillé dans les prochains travaux d'utiliser SARIMA avec les circuits ayant une saisonnalité de plus de quelques minutes. À moyen ou à long terme, il serait préférable d'utiliser la moyenne des données pour effectuer la prédiction de la qualité de performance.

Contrairement à ce qui est attendu, la prédiction des circuits calmes B et C n'était pas facile avec les méthodes étudiées. La prédiction est possible pour au plus 1 heure 45 minutes. L'agrégation

des données avec la moyenne est la meilleure prédiction pouvant être établie au-delà de cet horizon. Malgré leur bidirectionanlité et leur ressemblance, ces circuits ne partagent pas les mêmes hyperparamètres pour les réseaux de neurones. Chacun d'eux a été optimisé séparément.

Quant au circuit D, la capacité à prédire est bien différente, comparé aux autres circuits étudiés. Toutes les méthodes évaluées ont été capables de prédire la variable SNR jusqu'à un horizon de 24 heures, avec la supériorité du *stateful* LSTM. Toutefois, le modèle ARIMA a produit de bons résultats surtout en considérant sa simplicité en termes de calcul et temps requis pour l'optimisation des paramètres. La mise à jour de ce modèle durant la phase de test avec les anciennes observations est très bénéfique et réalisable en pratique.

Pour conclure cette comparaison, il n'y a pas un modèle commun capable de prédire la qualité de performance quelque soit le circuit optique, en utilisant uniquement le SNR comme *feature* et en ayant peu de données (12 mois). Cependant, le *stateful* LSTM et l'ARIMA pour les circuits non-saisonniers produisent des prédictions satisfaisantes. Il y a aussi des circuits prédictibles et d'autres plus difficile à prédire. Pour mieux étudier les performances du *stateful* LSTM, quelques expériences ont été réalisées. Les détails seront présentés dans la section 4.3.

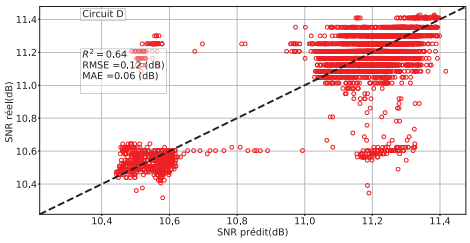
4.3 Analyse des performances de la méthode *stateful* LSTM

Les analyses effectuées se concentrent sur l'évolution des performances en fonction de l'horizon de prédiction, les bénéfices de la prédiction multivariée et celles de l'apprentissage par transfert.

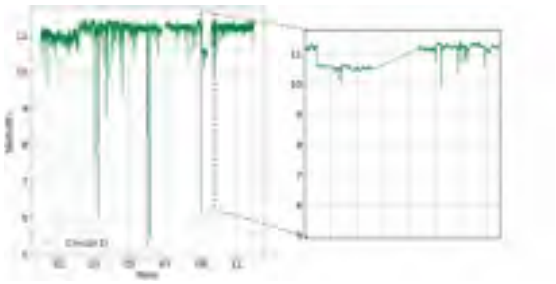
4.3.1 Analyse de l'effet de l'horizon de prédiction

Il est intéressant d'étudier l'évolution des performances de la méthode *stateful* LSTM à partir de l'horizon 24 heures. L'horizon testé se poursuit à 96 heures (soit 4 jours). Le circuit D était choisi pour cette analyse étant donné qu'il est le seul à être prédictible jusqu'à 24 heures. Le scénario utilisé est celui qui est décrit dans la section 4.2. Rappelons que les hyperparamètres de la méthode *stateful* LSTM pour le circuit D sont résumés dans la quatrième colonne du tableau 4.1. Commençons par visualiser SNR réel vs. SNR prédit pour un horizon de 24 heures

(figure 4.5a). Tel qu'illustré, le modèle prédictif n'est pas en mesure de prédire les sauts dans les valeurs de SNR qui se produisent durant le neuvième mois (figure 4.5b). Il prédit 11.2 et 10.5 dB approximativement alors que les valeurs réelles sont aux alentours de 10.6 et 11.2 dB respectivement. Ceci est attendu sachant que le réseau n'a pas vu ce genre de saut durant la phase d'entraînement. La figure 4.6 montre l'évolution des métriques de performance en fonction de l'horizon de prédiction (4 jours). Globalement les métriques RMSE, MAE augmentent en fonction de l'horizon de prédiction ce qui dégrade le pouvoir prédictif du modèle. Les valeurs de R^2 diminuent progressivement jusqu'à s'annuler à l'instant $t + 83$ heures. Quant à la direction de la moyenne d'erreur, le modèle a tendance à surestimer la qualité de performance. Finalement, la figure 4.6c représente la distribution de l'erreur absolue en fonction de l'horizon de prédiction. La ligne médiane dans chaque *boxplot* correspond à la médiane de la distribution et les moustaches correspondent à $1,5$ fois l'intervalle interquartile. Une stable distribution de l'erreur absolue avec une petite médiane est observée durant les premières 24 heures. Cette dernière distribution s'élargit de façon progressive jusqu'à l'horizon de 96 heures. En résumé, le modèle *stateful* LSTM perd son pouvoir prédictif à long terme, ce qui est prévu considérant le *feature* utilisé à l'entrée du modèle.

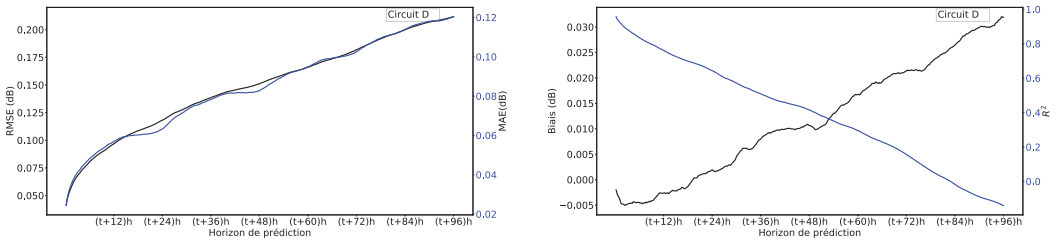


a) SNR réel vs. SNR prédit pour le circuit D, horizon =24 heures, méthode= *stateful* LSTM



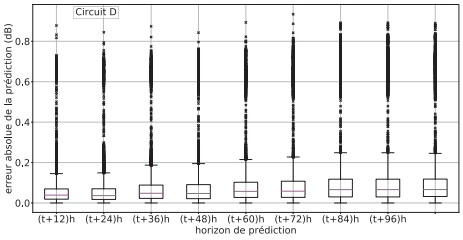
b) Représentation temporelle du circuit D

Figure 4.5 Pouvoir prédictif du circuit D



a) RMSE et MAE pour un horizon de 96 heures

b) Biais et R^2 pour un horizon de 96 heures



c) Distribution de l'erreur absolue de prédiction pour un horizon de 96 heures

Figure 4.6 Évolution des métriques de performance pour le circuit D en fonction de l'horizon de prédiction, méthode *stateful* LSTM

4.3.2 Analyse de l'effet de la prédiction multivariée

Dans la littérature, ajouter des attributs explicatifs pertinents à un modèle d'apprentissage automatique est généralement bénéfique pour la prédiction temporelle. Selon l'analyse effectuée dans la section 3.4, la puissance optique maximale reçue (OPR_MAX) est le seul attribut en relation avec la variable prédite SNR, et ce pour le circuit A. Dans cette partie, l'effet d'utiliser les valeurs historiques de SNR ainsi que celles de OPR_MAX à l'entrée de la méthode *stateful* LSTM est étudié. Le scénario est le même que dans la section 4.2. Notons bien que la différentiation n'est pas appliquée sur les données de OPR_MAX. En effet, en effectuant cette dernière opération, à laquelle ajouter la normalisation des données ne fait que diminuer la précision de la prédiction. Les hyperparamètres qui minimisent la racine carrée de l'erreur quadratique moyenne sur l'ensemble de validation seront utilisés durant la phase de test. Ces derniers ne sont pas aussi différents de ceux qui apparaissent dans la colonne une du tableau 4.1. Ainsi, seuls le nombre de neurones dans chaque couche cachée et le nombre d'époques ont

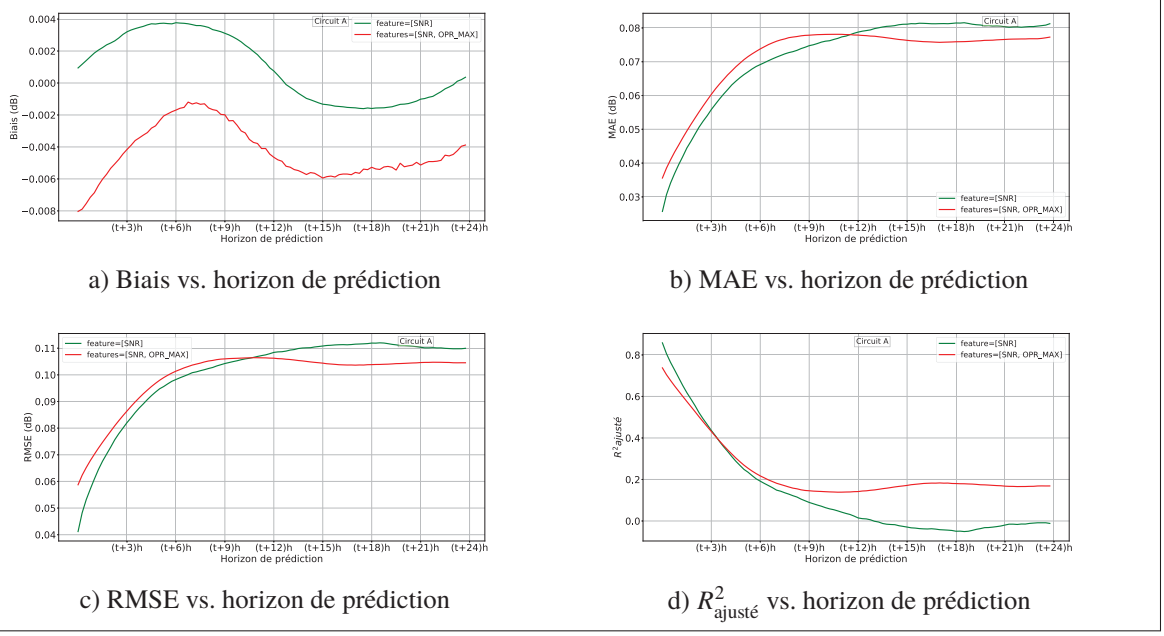
changé. En effet, le réseau de neurones a besoin de plus de neurones dans la deuxième couche cachée ($\mathbf{N_{LSTM}} = [100, 100]$). De plus, le modèle requiert moins d'époques qu'auparavant (($n_{epochs} = 75$). Comme expliqué dans la section 3.5, c'est le coefficient de détermination ajusté qui est utilisé pour la prédiction multivariée. Ce dernier, avec le biais, MAE et RMSE servent à évaluer les performances du modèle (figure 4.7). Le tableau 4.9 compare les médianes des métriques sur tous les instants prédits pour les deux prédictions, univariée et multivariée. La prédiction multivariée améliore légèrement la précision de la prédiction en matières de MAE, RMSE et $R^2_{\text{ajusté}}$. Cette évolution est plutôt observée à moyen terme (à partir de $t + 11$ heures) et continue jusqu'à $t + 24$ heures. Quant à la direction de la moyenne d'erreur, la prédiction multivariée sous-estime la qualité de prédiction, ce qui avantageux.

Tableau 4.9 Effet de la prédiction multivariée sur les performances du modèle *stateful* LSTM : circuit D

Modèle	MAE (dB)	RMSE (dB)	$R^2_{\text{ajusté}}$
***Stateful* LSTM, prédiction univariée**	0.07859	0.10818	0.18427
***Stateful* LSTM, prédiction multivariée**	0.07632	0.104227	0.01915

4.3.3 Analyse de l'effet de l'apprentissage par transfert

Le circuit D a été choisi pour tester l'impact de l'apprentissage par transfert sur les performances du modèle *stateful* LSTM. Le scénario pris en compte ici est le même que celui décrit dans la section 4.2. La première étape dans cette analyse est d'identifier les circuits sources en se basant sur la métrique mDTW (équation 3.17). La représentation temporelle de ces deux circuits sources (nommé Y et Z) ainsi que le circuit cible D est présentée dans la figure 4.8a. La valeur de mDTW calculée (mDTW=5521.26) pour ces circuits est la plus petite parmi les 150 circuits qui ont été évalués. D'étant plus, le circuit Y est plus similaire au circuit cible D que le circuit Z. Ainsi, toute la base de données du circuit Y sera utilisée en premier pour entraîner le modèle



a) Biais vs. horizon de prédiction

b) MAE vs. horizon de prédiction

c) RMSE vs. horizon de prédiction

d) $R^2_{\text{ajusté}}$ vs. horizon de prédiction

Figure 4.7 Évaluation des performances de la prédiction multivariée pour le circuit D, méthode *stateful* LSTM

stateful LSTM. Deux couches avec 100 unités chacune ont été implémentées pour cette analyse. Ensuite, la première couche du réseau de neurones a été gelée et le modèle a été réentraînée avec toutes la base de données du circuit Z. Finalement, la base de données d'entraînement du circuit cible D a été ajouté et les hyperparamètres ont été optimisés avec la base de données de validation. Les seuls hyperaparamètres qui ont changé par rapport à la colonne quatre du tableau 4.1 sont $n_{\text{batch}} = 128$, $n_{\text{epochs}} = 200$. La prédiction est effectuée sur un horizon de 24 heures. Les métriques de performance qui figurent dans le tableau 4.10 sont celles mesurées sur l'ensemble du test correspondant au circuit D. Les valeurs présentées sont les médianes calculées sur les différents instants de prédiction.

L'application de l'apprentissage par transfert a pu améliorer les prédictions temporelles. En effet, une amélioration d'environ 5% est observée dans les médianes des métriques. Une représentation de R^2 en fonction de l'horizon de prédiction est illustrée dans la figure 4.8b. Les valeurs de R^2 en utilisant le transfert par apprentissage sont meilleures que celles sans apprentissage par transfert sur tous les instants de prédiction. À $t + 24$ heures, une amélioration de 0.08 en termes

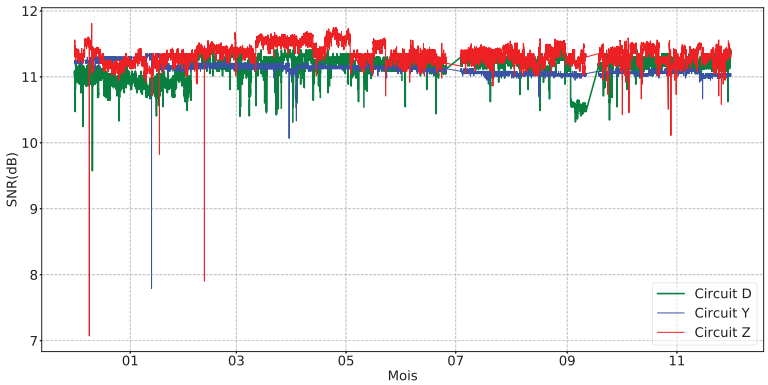
de R^2 est observée. En tout, l'application de l'apprentissage par transfert avec deux sources est avantageuse pour le circuit D. Cependant, ce n'est pas toutefois nécessaire de trouver des circuits sources qui ressemblent réellement au circuit cible dans notre base de données, tel est le cas pour le circuit A. En effet, les deux circuits qui lui étaient similaires en matière de mDTW n'ont fait qu'empirer les résultats de prédiction. Ceci laisse supposer qu'ils n'ont des patrons en communs avec le circuit A.

Tableau 4.10 Effet du l'apprentissage par transfert sur les performances du modèle *stateful* LSTM : circuit D

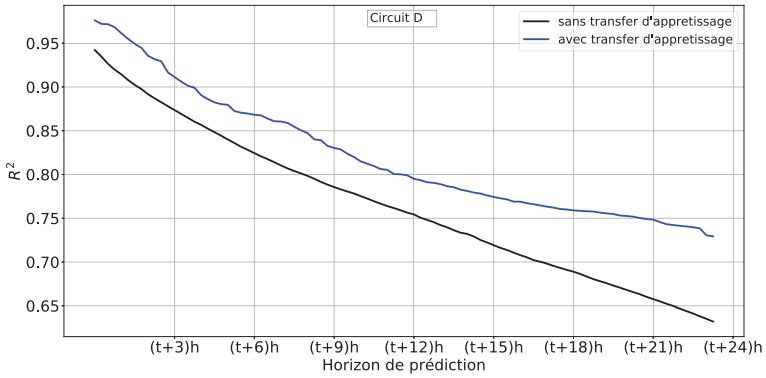
Modèle	MAE (dB)	RMSE (dB)	R^2
***Stateful* LSTM sans apprentissage par transfert**	0.0571	0.0969	0.7646
***Stateful* LSTM avec apprentissage par transfert**	0.0548	0.0891	0.7970

4.4 Sommaire des résultats

Le présent chapitre expose et analyse les résultats obtenus à partir des différents méthodes étudiées pour les circuits A, B, C et D. Cette analyse permet de déceler que les méthodes *stateful* LSTM et ARIMA (avec mise à jour durant le test) sont convenables pour la prédiction de la qualité de performance. La méthode ARIMA est en mesure de produire des résultats acceptables uniquement en présence de patrons linéaires dans les données pour les circuits non-saisonniers. Cependant, la méthode *stateful* LSTM n'a pas ces restrictions. Elle est capable de modéliser même les relations non-linéaires si le circuit est déjà prédictible. Ces résultats ne sont pas vérifiés avec tous les circuits étudiés. En effet, le pouvoir prédictif dépend du circuit et varie selon l'horizon de prédiction visé. Il y a deux circuits qui ne sont prédictibles qu'à très court terme : les circuits B et C. Malgré leurs plus importantes variations, la prédiction des circuits A et D est moins compliqué. Le modèle *stateful* LSTM requiert toutefois l'optimisation des hyperparamètres pour trouver une bonne configuration. Cette opération s'avère parfois difficile



a) Représentation temporelle du circuit D



b) R^2 vs. horizon de prédiction

Figure 4.8 Application de l'apprentissage par transfert pour le circuit D, méthode *stateful* LSTM

et requiert beaucoup de temps ainsi qu'une capacité de calcul importante. L'ARIMA avec mise à jour est une méthode assez simple, mais qui a produit des prédictions satisfaisantes.

Une série d'analyse a été effectuée avec la méthode *stateful* LSTM. Elle comprend l'évolution de l'horizon de prédiction, la prédiction multivariée et l'apprentissage par transfert. Selon les résultats obtenus, la méthode fournit de bonnes prédictions à long terme pour le circuit D. De plus, ajouter la puissance optique maximale reçue est avantageux pour le circuit A. Finalement, l'apprentissage par transfert est en mesure d'améliorer le pouvoir prédictif à condition que les circuits sources soient similaires au circuit cible. L'utilisation de l'apprentissage par transfert et de la prédiction multivariée constitue un avantage par rapport à la méthode ARIMA qui ne

supporte aucune de ces deux techniques. C'est plutôt la méthode ARIMAX (ARIMA *with Exogeneous Input*) qui pourrait être utilisée pour la prédiction multivariée.

CHAPITRE 5

CONCLUSION ET PERSPECTIVES

5.1 Conclusion

Dans ce projet, une étude sur la prédiction de la qualité de performance mesurée à travers le SNR est faite pour des circuits optiques opérationnels à grande capacité. L'horizon de prédiction évalué est de 24 heures. Le rapport s'est limité à quatre circuits de bout en bout avec des types d'installation diversifiés. Deux circuits sont calmes et bidirectionnels et deux autres appartiennent à des routes différentes et sont considérés comme dynamiques. Les données de monitoring utilisées dans ce projet couvrent uniquement 12 mois d'observations. Des réseaux de neurones de type *stateful* LSTM, *stateless* LSTM, *stateful* GRU et 1D-CNN sont analysés et comparés avec les méthodes ARIMA ou SARIMA dépendamment de l'existence ou non de la saisonnalité dans les circuits optiques. La comparaison est effectuée selon les métriques de performance suivantes : le biais, le MAE, RMSE et R^2. Une étape de préparation de données précède l'application des algorithmes d'apprentissage automatique. Elle comprend l'imputation des données manquantes, la gestion des valeurs aberrantes et la normalisation des données. Les valeurs aberrantes sont identifiées à l'aide d'une technique de mélange de Gausienne avec k composants. Un seuil de décision déterminé empiriquement permet de classifier une observation en normale ou aberrante. Ce seuil pourrait être spécifié de façon plus judicieuse en ayant des exemples étiquetés de valeurs aberrantes. De plus, les attributs explicatifs qui pourraient influencer la variable prédite SNR sont investigués pour un des circuits sélectionnés. L'étude de la corrélation, la multi-colinéarité et la mesure de l'importance des variables explicatives a montré que seule la puissance optique maximale reçue est en relation avec la qualité de performance prédite. Ce paramètre sera donc considéré dans la prédiction multivariée du SNR.

Les résultats de la prédiction univariée révèlent que ce ne sont pas tous les circuits étudiés qui contiennent des patrons prédictibles pour un horizon de 24 heures. L'horizon pouvant être prédit varie déjà d'un circuit à l'autre. De plus, un circuit calme n'est pas nécessairement

prédictible. Le contraire a d'ailleurs été observé dans cette étude. Globalement, la méthode *stateful* LSTM dépasse légèrement les autres méthodes évaluées. Cette dernière requiert par contre l'optimisation des hyperparamètres, une opération coûteuse en termes de temps et de ressources requises. Vient ensuite la méthode ARIMA avec mise à jour régulière durant la phase de test pour les circuits non saisonniers ayant des patrons linéaires. Malgré sa simplicité, elle produit des prédictions satisfaisantes. Pour les prochains travaux, il est conseillé d'éviter l'application de la méthode SARIMA pour les circuits ayant une saisonnalité de plus de quelques minutes. En effet, la détermination des paramètres optimaux est très lente avec un paramètre m aussi large, et les prédictions estimées sont médiocres. La quantité de données disponible pour ce projet constitue une limitation importante surtout avec une approche d'apprentissage automatique. Pour remédier partiellement à ce problème, l'apprentissage par transfert a été appliqué sur un des circuits sélectionnés, et ce à partir de deux sources de données. L'utilisation de cette technique avec la méthode *stateful* LSTM est avantageuse à condition de trouver deux circuits sources qui sont similaires au circuit cible de la prédiction. Finalement, l'ajout de la puissance maximale reçue comme *feature* à l'entrée de la méthode *stateful* LSTM a pu améliorer les prédictions surtout à moyen et long termes.

5.2 Perspectives

Plusieurs recommandations sont proposées dans la continuité de ce projet : elle comprend entre autres l'ajout d'autres attributs explicatifs dans nos modèles prédictifs, la prédiction en ligne de la qualité de performance, le développement d'un modèle unique pour tous les circuits optiques et la mise en place d'un entrepôt de données. Ces perspectives sont détaillées ci-dessous.

- pour la suite du projet, il nous semble important d'enrichir les modèles prédictifs avec de nouveaux attributs explicatifs. Intégrer des attributs pertinents à l'entrée des modèles pourrait améliorer considérablement la qualité de la prédiction sur un horizon plus large, ouvrant ainsi la voie à une maintenance proactive et à l'automatisation du réseau. Selon l'étude effectuée ici, la puissance maximale du signal optique reçue peut influencer la qualité de performance dans les réseaux optiques cohérents. Elle est donc prise en compte dans les

modèles évalués. D'autres attributs pourraient être investigués, à savoir le gain de chacun des amplificateurs utilisé dans le circuit optique. Les statistiques collectées sur les erreurs sont aussi des attributs explicatifs potentiels. De même, traiter et ajouter les fichiers *logs* des réparations serait potentiellement bénéfique. Il serait également intéressant d'inclure des statistiques descriptives du SNR dans les modèles. La moyenne, la variance et la tendance sont des exemples de ces statistiques. Les travaux de (Hyndman, Wang & Laptev, 2015) fournissent une liste non exhaustive de statistiques pouvant être appropriées pour des travaux futurs ;

- dans ce stade du projet, un modèle prédictif est entraîné pour chaque circuit étudié. Ce n'est pas la meilleure approche en pratique sachant qu'un opérateur peut avoir des millions de circuits dans son réseau optique et qu'un modèle d'apprentissage automatique est gourmand en termes de ressources requises. Toutefois, entraîner un seul modèle avec les données de tous les circuits, et ce en utilisant les attributs explicatifs précédemment mentionnés n'a pas produit des résultats compétitifs. Le modèle n'est pas en mesure de distinguer chacun des circuits. Actuellement, les attributs sont extraits et introduits manuellement dans les modèles. Cette méthode est fastidieuse et sujette aux erreurs. Les travaux de (Laptev, Yosinski, Li & Smyl, 2017) proposent une nouvelle architecture formant un modèle unique pour des séries temporelles provenant de sources différentes, grâce à un module d'extraction automatique des *features*. L'architecture est parfaitement applicable dans notre contexte. Ils utilisent un autoencodeur (*Autoencoder*) de type LSTM pour assurer cette dernière fonction. En effet, un tel réseau est capable d'apprendre automatiquement une bonne représentation des données temporelles. Une fois les vecteurs de *features* sont constitués, ils sont agrégés en utilisant les techniques d'ensembles. Le vecteur obtenu est ensuite concaténé avec les données temporelles du circuit étudié. Un réseau LSTM pourrait être utilisé pour assurer la prédiction temporelle ;
- une autre approche intéressante à explorer serait la prédiction en ligne (*online prediction*) de la qualité de performance dans les réseaux optiques opérationnels. Cette approche utilise des données observées en temps réel, contenant ainsi des valeurs aberrantes ou encore des changements dans les patrons de données (nommé *change points*). Rappelons que les

valeurs aberrantes sont des points ayant un comportement assez différent du reste de la série temporelle. Cependant, un point de changement indique que la distribution de données est significativement différente avant et après ce point. Les modèles d'apprentissage en ligne requièrent une gestion différente des anomalies, comparé à ce qui est proposé dans ce projet. En effet, ils devraient être capables de distinguer les valeurs aberrantes des points de changement. En cas d'apparition des valeurs aberrantes, le modèle ne devrait pas adapter son apprentissage, alors qu'il devrait le faire rapidement en identifiant un point de changement. Dans ce contexte, les travaux de (Guo, Xu, Yao, Chen, Aberer & Funaya, 2016) proposent un algorithme de gradient adaptatif pouvant être utilisé avec les réseaux de neurones récurrents. Ils pondèrent les poids du gradient en se basant sur une analyse des caractéristiques locales de la série temporelle. Les résultats obtenus sont très prometteurs ;

- les opérateurs sont en compétitions continues pour améliorer leurs qualités de services et satisfaire aux mieux les besoins de leurs clients. L'analyse des données de monitoring réelles est indispensable pour atteindre ces objectifs. Elle permet nécessairement une meilleure compréhension du dynamisme des performances dans les réseaux optiques déployés. Pour cette raison, les opérateurs collectent maintenant différents types de paramètres traduisant la qualité de transmission à des intervalles très fins (parfois des secondes). Au bout de quelques années, il y aura une immense quantité de données à stocker et à gérer. Ainsi, développer un entrepôt de données avec une architecture flexible et évolutive est primordial. La mise en place d'une architecture de type *Data vault* (Linstedt & Olschimke, 2015) pourrait être très avantageuse dans ce contexte. En effet, elle supporte de très larges bases de données, et ce en intégrant les outils de *Big Data* à savoir Hadoop, MangoDB et plusieurs autres bases de données NoSQL. Elle facilite aussi le processus d'ingestion de données à partir d'autres sources. De plus, elle permet d'ajouter facilement de nouvelles données sans perturber le design existant, etc.

ANNEXE I

FONCTIONS D'ACTIVATION

Dans un réseau neuronal, une fonction d'activation transforme la sortie de chaque neurone. Il s'agit d'une fonction non-linéaire appliquée à la sortie d'un neurone (figure 2.2). En quelques sorte, elle reproduit le potentiel d'activation qui se trouve dans les neurones biologiques du cerveau humain. Si un certain seuil de stimulation est atteint, la fonction d'activation décide de laisser passer ou bloquer l'information. La non-linéarité introduite par cette dernière favorise l'apprentissage de patrons complexes.

Le choix de la fonction d'activation est critique. En effet, elle a une influence directe sur l'apprentissage de la représentation et les performances du réseau de neurones. Un mauvais choix peut engendrer la disparition du gradient (Goodfellow *et al.*, 2016). Dans ce cas, le réseau n'est pas en mesure d'optimiser ses poids et a tendance à stagner.

Plusieurs fonctions d'activations sont connues pour les réseaux de neurones. Les fonctions sigmoïde, tangeante hyperbolique (notée tanh) et ReLu (*Rectified Linear Unit*) sont celles évaluées dans le cadre de cette recherche. Elle sont détaillées ci-dessous :

Fonction sigmoïde :

La fonction d'activation sigmoïde (notée σ) convertit la somme pondérée du neurone en une valeur comprise entre 0 et 1. Elle est idéalement utilisée pour les modèles voulant prédire les sorties comme étant des lois probabilistes. Comme illustrée à la figure I-1, si x tend vers une grande valeur positive, la fonction sigmoïde tend vers 1. Dans le cas contraire, σ s'approche de la valeur 0. Ceci implique une faible mise à jour du gradient lorsque cette dernière fonction est confiante dans sa prédiction. La convergence est ainsi ralentie.

$$\sigma(x) = \frac{1}{1 + \exp(-x)} \tag{A I-1}$$

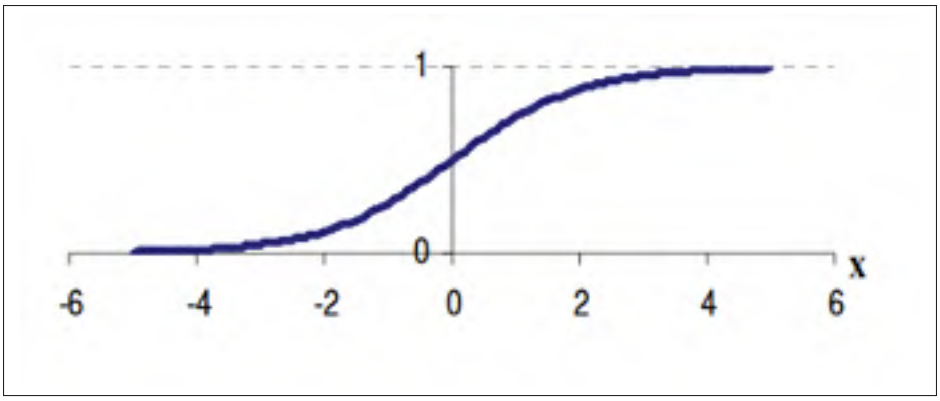


Figure-A I-1 Représentation de la fonction sigmoïde
Tirée de Karlik & Olgac (2011)

Fonction tangeante hyperbolique :

La fonction d'activation tangeante hyperbolique (notée tanh) est très semblable à la sigmoïde sauf qu'elle produit des valeurs entre -1 et 1. Elle est donnée par l'équation A I-2. La tanh est préférable par rapport au sigmoïde du fait qu'elle est centrée autour de zéro, ce qui accélère l'apprentissage (Le Cun, Kanter & Solla, 1991) comparée à σ. Cependant, tout comme la fonction sigmoïde, la tangeante hyperbolique peut conduire à la disparition du gradient.

$$\tanh(x) = \frac{\sinh(x)}{\cosh(x)} = \frac{\exp(x) - \exp(-x)}{\exp(x) + \exp(-x)} \qquad \text{(A I-2)}$$

Fonction ReLu :

La fonction ReLU a été initialement introduite par (Nair & Hinton, 2010). Elle est fréquemment utilisée dans les couches cachées des réseaux de neurones. Comme le montre l'équation suivante, la fonction attribue zéro aux valeurs strictement négatives. Cependant, elle laisse inchangée celles positives. De ce fait, les neurones ne sont pas tous activés au même temps. Certains neurones peuvent aussi ne jamais être activés. La popularité de cette fonction provient également de sa simplicité. En effet, le calcul de la dérivée est très peu coûteux ce qui accélère considérablement

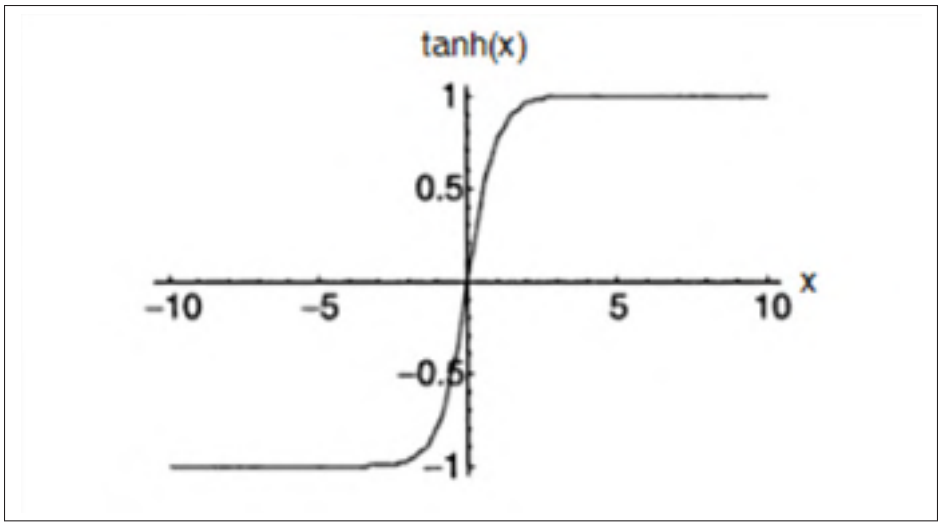


Figure-A I-2 Représentation de la fonction tangeante hyperbolique
Tirée de Karlik & Olgac (2011)

la phase d'apprentissage.

$$\mathrm{ReLu}(x) = \begin{cases} 0, & \text{pour } x < 0 \\ x, & \text{pour } x \geq 0 \end{cases} \qquad \mathrm{ReLu}'(x) = \begin{cases} 0, & \text{pour } x < 0 \\ 1, & \text{pour } x \geq 0 \end{cases} \tag{A I-3}$$

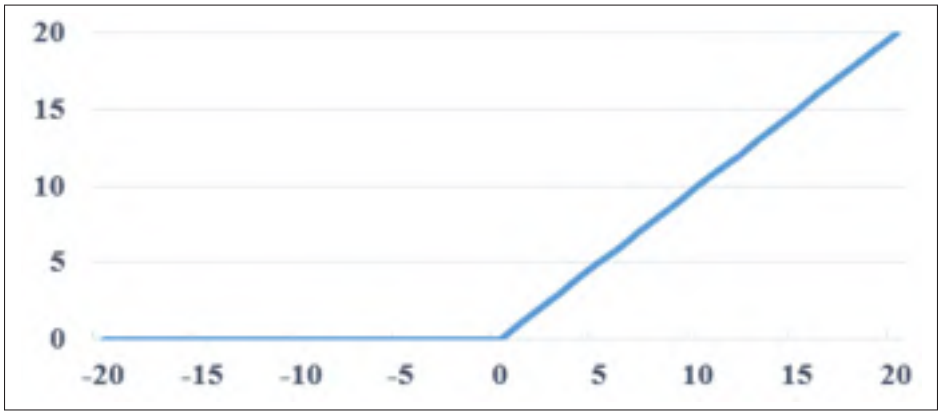


Figure-A I-3 Représentation de la ReLU

ANNEXE II

DEEP LEARNING FOR MULTI-STEP PERFORMANCE PREDICTION IN OPERATIONAL OPTICAL NETWORKS

Ameni Mezni[1], Doug Charlton[2], Christine Tremblay [1], Christian Desrosiers [1]

[1] École de Technologie Supérieure, Network Technology Lab,
1100 Notre-Dame Ouest, Montréal, Québec, Canada H3C 1K3
[2]Ciena Corp. Ottawa, ON, Canada

Article publié dans la conférence « CLEO Laser Science to Photonic Applications » en mai 2020.

Abstract : We propose a 1D Convolutional Neural Network to predict the performance of a lightpath using field bit error rate data and we evaluate the model robustness over forecast horizons up to 24 hours. © 2020 The Author(s)

1. Introduction

The increasing popularity of video streaming, 5G mobile extensive deployment, Internet of Things and data centers inter-connectivity have led to a tremendous amount of data transported over the Internet. Optical network operators are facing significant challenges to meet the capacity requirements in cost-efficient ways. To guarantee an error free transmission, a static system margin is allocated to account for factors such as equipment degradation, fiber aging and power fluctuations. This residual margin hinders the operator's efforts to fully exploit the available network bandwidth. Squeezing the security margin to a near-zero level may help maximize the delivered bandwidth, therefore a rigorous understanding of the network daily behavior is essential. This behavior can be learned by collecting parameters regarding signal quality such as optical power, bit error rate (BER), Q-factor and optical signal-to-noise ratio (OSNR). Performance Monitoring (PM) and performance prediction are promising solutions for this purpose (Dong *et al.*, 2015). Recent research has shown that a Deep Neural Network (DNN) trained with horizontal in-phase (HI), horizontal quadrature (HQ), vertical in-phase (VI) and vertical quadrature (VQ) simulated data, asynchronously sampled by a coherent receiver can

effectively estimate the optical SNR (OSNR) (Tanimura *et al.*, 2016,1). Despite this result, so far, no work has studied the multi-step SNR prediction using field PM data. In this paper, we bridge this gap by demonstrating the effectiveness of a Convolutional Neural Network (CNN) to model and predict SNR changes in a lightpath deployed on an aerial link during the next 24 hours.

2. Lightpath description

The PM dataset used in this study was collected at 15-minute intervals over 12 months in the production network of a large US carrier. The monitored lightpath uses a polarization multiplexed-quadrature phase shift keying modulation format (PM-QPSK) at 100 Gb/s and is deployed on a 350 km aerial optically amplified link including reconfigurable optical add-drop multiplexers (ROADMs). SNR observations are computed from the pre-forward error correction (pre-FEC) BER. During the observation period, 33, 463 BER samples were collected. The SNR varied between 6.432 and 13.052 dB with a median of 12.976 dB and a coefficient of variation of 0.004%.

3. 1D-CNN for performance prediction

3.1 Data preprocessing

To determine if the given time series is stationary, we applied an augmented Dickey-Fuller (ADF) test and a Kwiatkowski-Phillips-Schmidt-Shin (KPSS) test which respectively determine the presence of unit root and trend stationarity. Results showed the data to be difference-stationary ($ADF = -8.79$, $KPSS = 8.29$ and p-values of 0 and 0.01, respectively), hence a first order differencing was applied as preprocessing. A second application of the two tests on the transformed series confirmed its stationarity. To limit the impact of outliers, we then normalized transformed samples by removing their median and scaling them by the interquartile range before feeding them to the neural network.

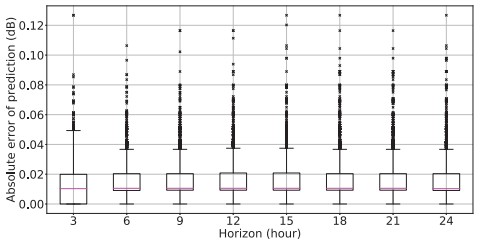
3.2 CNN architecture and parameter setup

CNNs have been extensively used in various computer vision tasks and, more recently, for the forecasting of time series data. Intuitively, applying a 1D-CNN for time series prediction consists in learning filters that can represent recurrent patterns in data and using them to predict future values. In this work, we tested multiple CNN architectures obtained the best results with a network composed of two convolution layers, each one followed by a pooling layer, and two fully-connected (FC) layers. The input to the network corresponds to a window containing the three last days of SNR data (3 days $\times$96 samples per day = 288 samples). Both convolution layers have 8 filters with a kernel size of 10 and stride of 2. Max pooling layers have a window size of 2. After the second pooling layer, the feature map is flattened to a vector and passed to the FC layers having 192 and 96 output neurons, respectively. The rectified linear unit (ReLU) was employed as activation function for all convolution and FC layers. During training, we set batch size to 64 and the number of epochs to 200. We also used dropout to reduce overfitting, where 20% of units in FC layers are randomly shut down at each training iteration. To learn the network parameters, we considered the mean square error (MSE) as loss function and employed the Adam optimizer (Kingma & Ba, 2014) with an initial learning rate of 0.008. We performed a 3-fold time series cross-validation technique to average the model performance metrics. The training examples were divided into a ratio of 80 : 20 for the purposes of training and validation respectively. The loss computed on validation examples was used to fine-tune the method's hyperparameters. Our proposed model was evaluated using three metrics : root mean squared error (RMSE), mean absolute percentage error (MAPE) and coefficient of determination (R^2). Whereas the RMSE corresponds to the standard deviation of the difference between predicted and observed SNR data, the MAPE measures the mean ratio between prediction errors and values to predict. In both cases, a value closer to 0 corresponds to a higher prediction accuracy. On the other hand, R^2 represents the proportion of the variance in the dependent variable that is predictable from the independent variable. A value closer to 1 implies a better fitting model.

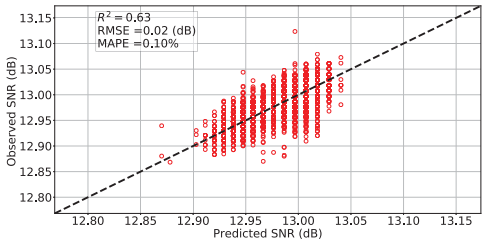
4. Results and discussion

The prediction accuracy of the trained model was evaluated on the test set. Metrics presented in this section are averaged over the 3-fold time series cross validation sets which have comparable

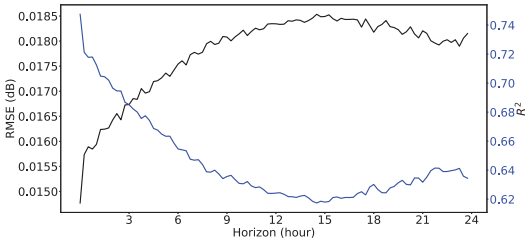
results. We assessed the model performance for forecast horizons ranging from a single step (15 minutes after the last observation) to 24 hours ahead. The distribution of absolute errors for tested forecast horizons is presented in figure II-1a, where middle line in each boxplot is the distribution median and whiskers correspond to 1.5 times the interquartile range (IQR). We observe a stable distribution of absolute errors with small median across different horizons. Figure II-1b shows the predicted SNR as function of the observed SNR after 24 hours. The CNN model was able to learn the temporal change in the data and its prediction can explain 63% of the variance in the observed SNR. The variation of RMSE and R^2 for each horizon is further detailed in figure II-1c. We see that our 1D-CNN model yields a high accuracy for short-term predictions and that, as expected, accuracy reduces for longer-term predictions. Interestingly, the mean prediction error decreases near a forecast horizon of 24 hours, which could be explained by a 24-hour periodicity in the data.



(a) Absolute error of prediction vs. forecast horizon



(b) Observed vs. predicted SNR heures



(c) RMSE and R^2 vs. forecast horizon

Figure-A II-1 Performance evaluation for the 1D-CNN method

5. Conclusion and outlook

This work proposed a new approach for multi-step performance prediction in operational optical networks using a 1D-CNN. Our model shows promising results : it can capture the temporal change in SNR data and predict it correctly 24 hours ahead. As a future work, we will compare the performance of our 1D-CNN to other architectures such as the Long Short-Term Memory (LSTM) network, on both quiet and dynamic lightpaths. We also plan to incorporate additional signal quality parameters as input and will investigate transfer learning for improving prediction accuracy.

6. Acknowledgments

This work was supported by the Natural Sciences and Engineering Research Council (NSERC) of Canada under grant CRDPJ 488332 – 15.

REFERENCES

Dong, Z., Khan, F. N., Sui, Q., Zhong, K., Lu, C. & Lau, A. P. T. (2015). Optical performance monitoring : A review of current and future technologies. *Journal of Lightwave Technology*, 34(2), 525–543.

Kingma, D. P. & Ba, J. (2014). Adam : A method for stochastic optimization. *arXiv preprint arXiv :1412.6980*.

Tanimura, T., Hoshida, T., Rasmussen, J. C., Suzuki, M. & Morikawa, H. (2016). OSNR monitoring by deep neural networks trained with asynchronously sampled data. *2016 21st OptoElectronics and Communications Conference (OECC) held jointly with 2016 International Conference on Photonics in Switching (PS)*, pp. 1–3.

Tanimura, T., Hoshida, T., Kato, T., Watanabe, S. & Morikawa, H. (2018). Data-analytics-based optical performance monitoring technique for optical transport networks. *Optical Fiber Communication Conference*, pp. Tu3E–3.

RÉFÉRENCES

Abadi, M., Barham, P., Chen, J., Chen, Z., Davis, A., Dean, J., Devin, M., Ghemawat, S., Irving, G., Isard, M. et al. (2016). Tensorflow : A system for large-scale machine learning. *12th {USENIX} symposium on operating systems design and implementation ({OSDI} 16)*, pp. 265–283.

Aladin, S. & Tremblay, C. (2018). Cognitive tool for estimating the QoT of new lightpaths. *Optical Fiber Communication Conference*, pp. M3A–3.

Aladin, S., Allogba, S., Tran, A. V. S. & Tremblay, C. (2020a). Recurrent neural networks for short-term forecast of lightpath performance. *Optical Fiber Communication Conference*, pp. W2A–24.

Aladin, S., Tran, A. V. S., Allogba, S. & Tremblay, C. (2020b). Quality of transmission estimation and short-term performance forecast of lightpaths. *Journal of Lightwave Technology*, 38(10), 2806–2813.

Alam, S. J., Alam, M. R., Hu, G. & Mehrab, M. Z. (2011). Bit error rate optimization in fiber optic communications. *International Journal of Machine Learning and Computing*, 1(5), 435.

Allogba, S. & Tremblay, C. (2018). K-Nearest neighbors classifier for field bit error rate data. *2018 Asia Communications and Photonics Conference (ACP)*, pp. 1–3.

Amaral, T., Silva, L. M., Alexandre, L. A., Kandaswamy, C., de Sá, J. M. & Santos, J. M. (2014). Transfer learning using rotated image data to improve deep neural network performance. *International Conference Image Analysis and Recognition*, pp. 290–300.

Bayer, J., Wierstra, D., Togelius, J. & Schmidhuber, J. (2009). Evolving memory cell structures for sequence learning. *International Conference on Artificial Neural Networks*, pp. 755–764.

Bergstra, J. & Bengio, Y. (2012). Random search for hyper-parameter optimization. *Journal of machine learning research*, 13(Feb), 281–305.

Berndt, D. J. & Clifford, J. (1994). Using dynamic time warping to find patterns in time series. *KDD workshop*, 10(16), 359–370.

Canova, F. & Hansen, B. E. (1995). Are seasonal patterns constant over time ? A test for seasonal stability. *Journal of Business & Economic Statistics*, 13(3), 237–252.

Chan, C. C. (2010). *Optical performance monitoring : advanced techniques for next-generation photonic networks*. Academic Press.

Cho, K., Van Merriënboer, B., Bahdanau, D. & Bengio, Y. (2014). On the properties of neural machine translation : Encoder-decoder approaches. *arXiv preprint arXiv :1409.1259*.

Chollet, F. et al. (2015). Keras. GitHub.

Chung, J., Gulcehre, C., Cho, K. & Bengio, Y. (2014). Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv preprint arXiv :1412.3555*.

Dong, Z., Khan, F. N., Sui, Q., Zhong, K., Lu, C. & Lau, A. P. T. (2015). Optical performance monitoring : A review of current and future technologies. *Journal of Lightwave Technology*, 34(2), 525–543.

Dunsmuir, W. & Robinson, P. (1981). Estimation of time series models in the presence of missing data. *Journal of the American Statistical Association*, 76(375), 560–568.

Elman, J. L. (1990). Finding structure in time. *Cognitive science*, 14(2), 179–211.

Frenzel, L. E. (2016). Handbook of Serial Communications Interfaces. *Elsevier Inc.*

Freude, W., Schmogrow, R., Nebendahl, B., Winter, M., Josten, A., Hillerkuss, D., Koenig, S., Meyer, J., Dreschmann, M., Huebner, M. et al. (2012). Quality metrics for optical signals : Eye diagram, Q-factor, OSNR, EVM and BER. *2012 14th International Conference on Transparent Optical Networks (ICTON)*, pp. 1–4.

Gers, F. A., Schmidhuber, J. & Cummins, F. (1999). Learning to forget : Continual prediction with LSTM.

Goodfellow, I., Bengio, Y. & Courville, A. (2016). *Deep learning*. MIT press.

Graves, A., Fernández, S. & Schmidhuber, J. (2005). Bidirectional LSTM networks for improved phoneme classification and recognition. *International Conference on Artificial Neural Networks*, pp. 799–804.

Guo, T., Xu, Z., Yao, X., Chen, H., Aberer, K. & Funaya, K. (2016). Robust online time series prediction with recurrent neural networks. *2016 IEEE International Conference on Data Science and Advanced Analytics (DSAA)*, pp. 816–825.

Hadri, K. & Rao, Y. (2009). KPSS test and model misspecifications. *Applied Economics Letters*, 16(12), 1187–1190.

He, Q.-Q., Pang, P. C.-I. & Si, Y.-W. (2019). Transfer Learning for Financial Time Series Forecasting. *Pacific Rim International Conference on Artificial Intelligence*, pp. 24–36.

Hochreiter, S. & Schmidhuber, J. (1997). Long short-term memory. *Neural computation*, 9(8), 1735–1780.

Hochreiter, S., Bengio, Y., Frasconi, P., Schmidhuber, J. et al. (2001). Gradient flow in recurrent nets : the difficulty of learning long-term dependencies. A field guide to dynamical recurrent neural networks. IEEE Press.

Hyndman, R. & Athanasopoulos, G. (2018). Forecasting : principles and practice, OTexts : Melbourne. *Australia. OTexts. com/fpp2. Accessed on*, 18(05), 2019.

Hyndman, R. J., Wang, E. & Laptev, N. (2015). Large-scale unusual time series detection. *2015 IEEE international conference on data mining workshop (ICDMW)*, pp. 1616–1619.

Jordan, M. I. (1997). Serial order : A parallel distributed processing approach. Dans *Advances in psychology* (vol. 121, pp. 471–495). Elsevier.

Kao, K. & Hockham, G. A. (1966). Dielectric-fibre surface waveguides for optical frequencies. *Proceedings of the Institution of Electrical Engineers*, 113(7), 1151–1158.

Karlik, B. & Olgac, A. V. (2011). Performance analysis of various activation functions in generalized MLP architectures of neural networks. *International Journal of Artificial Intelligence and Expert Systems*, 1(4), 111–122.

Karlsson, M. & Agrell, E. (2009). Which is the most power-efficient modulation format in optical links ? *Optics express*, 17(13), 10814–10819.

Keiser, G. (2010). Optical fiber communications.

Keogh, E. J. & Pazzani, M. J. (2001). Derivative dynamic time warping. *Proceedings of the 2001 SIAM international conference on data mining*, pp. 1–11.

Kingma, D. P. & Ba, J. (2014). Adam : A method for stochastic optimization. *arXiv preprint arXiv :1412.6980*.

Laptev, N., Yosinski, J., Li, L. E. & Smyl, S. (2017). Time-series extreme event forecasting with neural networks at uber. *International Conference on Machine Learning*, 34, 1–5.

Le Cun, Y., Kanter, I. & Solla, S. A. (1991). Eigenvalues of covariance matrices : Application to neural-network learning. *Physical Review Letters*, 66(18), 2396.

Lecoy, P. (2015). La fibre optique dans les bâtiments résidentiels et professionnels.

LeCun, Y. et al. (1989). Generalization and network design strategies. *Connectionism in perspective*, 19, 143–155.

Lim, B. & Zohren, S. (2020). Time Series Forecasting With Deep Learning : A Survey. *arXiv preprint arXiv :2004.13408*.

Linstedt, D. & Olschimke, M. (2015). *Building a scalable data warehouse with data vault 2.0*. Morgan Kaufmann.

Mezni, A., Charlton, D., Tremblay, C. & Desrosiers, C. (2020). Deep Learning for Multi-Step Performance Prediction in Operational Optical Networks. *CLEO - Laser Science to Photonic Applications*.

Mitchell, T. M. et al. (1997). Machine learning. McGraw-hill New York.

Nair, V. & Hinton, G. E. (2010). Rectified linear units improve restricted boltzmann machines. *ICML*.

Qian, N. (1999). On the momentum term in gradient descent learning algorithms. *Neural networks*, 12(1), 145–151.

Rangapuram, S. S., Seeger, M. W., Gasthaus, J., Stella, L., Wang, Y. & Januschowski, T. (2018). Deep state space models for time series forecasting. *Advances in neural information processing systems*, pp. 7785–7794.

Robbins, H. & Monro, S. (1951). A stochastic approximation method. *The annals of mathematical statistics*, 400–407.

Roberts, K., O'Sullivan, M., Wu, K.-T., Sun, H., Awadalla, A., Krause, D. J. & Laperle, C. (2009). Performance of dual-polarization QPSK for optical transport systems. *Journal of lightwave technology*, 27(16), 3546–3559.

Rosenstein, M. T., Marx, Z., Kaelbling, L. P. & Dietterich, T. G. (2005). To transfer or not to transfer. *NIPS 2005 workshop on transfer learning*, 898, 1–4.

Ruiz, M., Fresi, F., Vela, A. P., Meloni, G., Sambo, N., Cugini, F., Poti, L., Velasco, L. & Castoldi, P. (2016). Service-triggered failure identification/localization through monitoring of multiple parameters. *ECOC 2016 ; 42nd European Conference on Optical Communication*, pp. 1–3.

Rumelhart, D. E., Hinton, G. E. & Williams, R. J. (1986). Learning representations by back-propagating errors. *nature*, 323(6088), 533–536.

Salinas, D., Flunkert, V., Gasthaus, J. & Januschowski, T. (2019). DeepAR : Probabilistic forecasting with autoregressive recurrent networks. *International Journal of Forecasting*.

Savory, S. J. (2013). Digital signal processing for coherent optical communication systems. *OptoElectronics and Communications Conference and Photonics in Switching*, pp. TuR3_1.

Schaul, T., Antonoglou, I. & Silver, D. (2013). Unit tests for stochastic optimization. *arXiv preprint arXiv :1312.6055*.

Semeniuta, S., Severyn, A. & Barth, E. (2016). Recurrent dropout without memory loss. *arXiv preprint arXiv :1603.05118*.

Singh, S. P. & Singh, N. (2007). Nonlinear effects in optical fibers : origin, management and applications. *progress in Electromagnetics Research*, 73, 249–275.

Slutzky, E. (1937). The summation of random causes as the source of cyclic processes. *Econometrica : Journal of the Econometric Society*, 105–146.

Smith, T. G. et al. (2017). pmdarima : Arima estimators for python. MIT, USA.

Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I. & Salakhutdinov, R. (2014). Dropout : a simple way to prevent neural networks from overfitting. *The journal of machine learning research*, 15(1), 1929–1958.

Tanimura, T., Hoshida, T., Rasmussen, J. C., Suzuki, M. & Morikawa, H. (2016). OSNR monitoring by deep neural networks trained with asynchronously sampled data. *2016 21st OptoElectronics and Communications Conference (OECC) held jointly with 2016 International Conference on Photonics in Switching (PS)*, pp. 1–3.

Tanimura, T., Hoshida, T., Kato, T., Watanabe, S. & Morikawa, H. (2018). Data-analytics-based optical performance monitoring technique for optical transport networks. *Optical Fiber Communication Conference*, pp. Tu3E–3.

Tremblay, C. (2018). ELE772 : Communications optiques [Notes de cours]. Repéré à https://ena.etsmtl.ca/pluginfile.php/545127/mod_resource/content/1/ELE772%20H18%20Cours%201_v1%202_Final.pdf.

Tremblay, C., Allogba, S. & Aladin, S. (2019). Quality of transmission estimation and performance prediction of lightpaths using machine learning.

Vela, A. P., Ruiz, M., Fresi, F., Sambo, N., Cugini, F., Meloni, G., Potì, L., Velasco, L. & Castoldi, P. (2017). BER degradation detection and failure identification in elastic optical networks. *Journal of Lightwave Technology*, 35(21), 4595–4604.

Vu, N. T., Imseng, D., Povey, D., Motlicek, P., Schultz, T. & Bourlard, H. (2014). Multilingual deep neural network based acoustic modeling for rapid language adaptation. *2014 IEEE international conference on acoustics, speech and signal processing (ICASSP)*, pp. 7639–7643.

Wang, Z., Zhang, M., Wang, D., Song, C., Liu, M., Li, J., Lou, L. & Liu, Z. (2017). Failure prediction using machine learning and time series in optical network. *Optics Express*, 25(16), 18553–18565.

Yaméogo, B. L. M. (2017). *ANALYSE DE TENDANCES ET PRÉDICTION DE PANNES DANS LES RÉSEAUX OPTIQUES COHÉRENTS*. Rapport de DGA-1031, ÉCOLE DE TECHNOLOGIE SUPÉRIEURE, Montréal.

Yosinski, J., Clune, J., Bengio, Y. & Lipson, H. (2014). How transferable are features in deep neural networks ? *Advances in neural information processing systems*, pp. 3320–3328.

Young, T., Hazarika, D., Poria, S. & Cambria, E. (2018). Recent trends in deep learning based natural language processing. *ieee Computational intelligenCe magazine*, 13(3), 55–75.