

Quality of Transmission Estimation and Short-Term Performance Forecast of Lightpaths

Sandra Aladin, Anh Vu Stephan Tran, Stéphanie Allogba and Christine Tremblay

Abstract— With ever-increasing traffic, the need more dynamic, flexible and autonomous optical networks is more important than ever. The availability of performance monitoring data makes it possible to leverage machine learning (ML) for fast quality of transmission (QoT) estimation and performance prediction of lightpaths in complex optical networks. In this work, we explore classifiers based on support vector machine (SVM) and artificial neural network (ANN) for QoT estimation of unestablished lightpaths. Using a synthetic knowledge base (KB), the classification accuracy of the ANN and SVM models decreased from 99%, with a complete feature set, to 85.03% and 88.52%, respectively, with a reduced feature set. We also propose a Long Short-Term Memory (LSTM), an Encoder-Decoder LSTM and a Gated Recurrent Unit (GRU) models, trained with 13-month field performance data, for lightpath signal-to-noise (SNR) prediction over forecast horizons up to 4 days. Positive R^2 values combined with low (< 0.285 dB) root mean square error (RMSE) indicated that the GRU model achieved slightly better predictions than the naive method for forecast horizons ranging from 1 to 96 hours, whereas the LSTM performed better over 24 to 96-hour forecast horizons. The Encoder-Decoder LSTM model achieved the lowest R^2 and the highest RMSE values (0.296 dB). Additional input data will be needed to improve the prediction accuracy of the LSTM and GRU models trained with single lightpath data.

Index Terms— Artificial Neural Network, Gated Recurrent Unit, Long Short-Term Memory, Machine Learning, Performance Prediction, Quality of Transmission, Recurrent Neural Networks, Support Vector Machine.

I. INTRODUCTION

THE continuous traffic increase over the years has led to the deployment of wavelength-division multiplexing (WDM) optical systems with ever-increasing data rates, capacity and flexibility. Video and cloud applications, as well as emerging 5G and Internet of Things (IoT) applications call for even higher traffic volumes, heterogeneity and dynamicity in optical networks. This will make the potential impact of performance degradation and failure at the link and network

levels more severe and the need for flexible and autonomous network management more important than ever. The availability of transponders with monitoring capabilities makes it possible to leverage the potential of machine learning (ML) to design and manage increasingly heterogeneous, dynamic and complex optical networks in a software defined network (SDN) context.

ML algorithms for optical networking applications have been explored in the last years [1]. ML-based methods such as K-nearest neighbours (K-NN), support vector machine (SVM) and random forest (RF) have been proposed for estimating the quality of transmission (QoT) of unestablished lightpaths [2]. A comparative study of ML-based lightpath classifiers realized with a synthetic knowledge base (KB) of 25,600 bit error rate (BER) instances generated using the Gaussian noise model has shown that SVM outperforms RF and K-NN in terms of class prediction's accuracy [2]. A ML physical layer model (ML-PLM) trained with physical layer parameters and a ML model (ML-M) using monitoring metrics combined with analytically generated link-based features were found to achieve a very good accuracy with a relatively small training database [3]. ML-based methods have also been explored for predicting performance metrics such as the BER, signal-to-noise ratio (SNR) or Q-factor for deployed lightpaths. Performance prediction would allow network operators to respond proactively to performance degradations or potential failures in optical networks. ML has also been proposed for predicting traffic matrix, optical path performance and equipment failure [4, 5]. A long short-term memory (LSTM) algorithm has been proposed to predict reconfigurable optical add-drop multiplexer (ROADM) network resource requirements 30 minutes in advance [6]. The proposed LSTM and Gated Recurrent Unit (GRU) models trained with field lightpath data for short-term SNR forecasting have produced promising results [7].

In [8], we obtained a 99.38% accuracy in estimating the QoT of unestablished lightpaths by using a SVM QoT estimator trained with 30,270 BER synthetic data instances. We also proposed a LSTM model, trained with 13 months of field BER data for one lightpath, for predicting the SNR of the lightpath over forecast horizons up to 24 hours. In this paper, we extend the work presented in [8] by investigating a new QoT estimator based on artificial neural network (ANN) and two new SNR prediction models based on LSTM and Gated Recurrent Unit (GRU). The two use cases considered in this study are shown in Fig. 1. In the first use case, QoT estimation of unestablished lightpaths using SVM and ANN, as well as a synthetic BER database, is performed. In the second use case, LSTM, Encoder-Decoder LSTM and GRU models trained

Manuscript submitted on December 1, 2019.

This work was supported by the Natural Sciences and Engineering Research Council (NSERC) of Canada under grant RGPIN-2019-03972. The authors thank Thomas Tam and Tao Zhang from CANARIE for field performance data.

S. Aladin, A. V. S. Tran, S. Allogba and C. Tremblay are with the Network Technology Lab, Department of Electrical Engineering, École de technologie supérieure, Montréal, Qc, Canada H3C 1K3 (e-mail: christine.tremblay@etsmtl.ca).

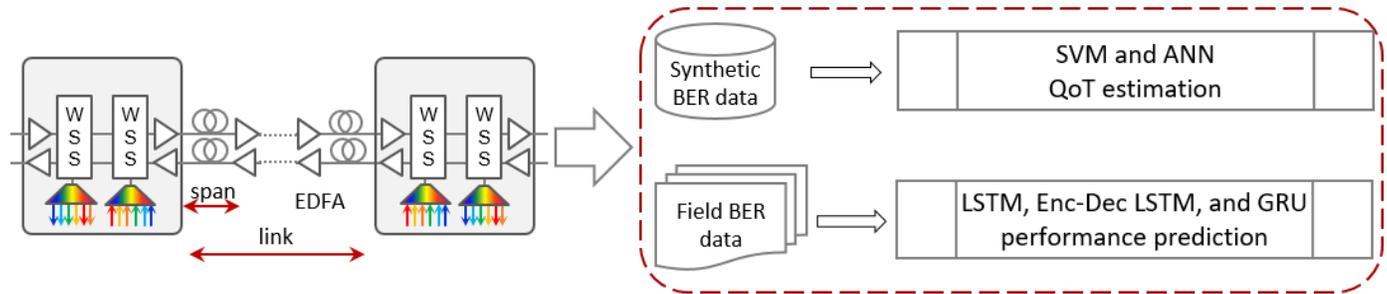


Fig. 1. ML in optical networking: overview of the two use cases considered in this study (WSS: wavelength selective switch; EDFA: Erbium-doped fiber amplifier; Enc-Dec: Encoder-Decoder)

Table I Optical system and link parameters in the KB

Parameter	Value
Link length	80 to 7500 km
Span length	80, 100, 120, 150 km
Number of spans	1 to 50
Modulation format	PM-BPSK, PM-QPSK, PM-16QAM, PM-64QAM
Channel power	-10 to 5 dBm
Data rate	40, 50, 100 Gb/s

Polarization-Multiplexed (PM); Binary Phase-Shift-Keying (BPSK); Quadrature Phase Shift Keying (QPSK); Quadrature Amplitude Modulation (QAM)

Table II Fixed parameters values

Parameter	Value
Fiber attenuation coefficient	0.2 dB/km
Dispersion coefficient	-21 (ps) ² /km
Nonlinear coefficient	1.3 W ⁻¹ km ⁻¹
Amplifier noise figure	5 dB
Central frequency	193.1 THz
Noise bandwidth	32 GHz
Reference optical bandwidth	12.5 GHz
Number of channels	9
Baud rate	32 Gbaud

with historical field performance data for one lightpath are used to predict the SNR for one lightpath over time horizons up to 4 days.

The remainder of the document is organized as follows. The QoT estimation and SNR forecast use cases are presented in Section 2 and Section 3, respectively. Discussion and concluding remarks are included in Section 4.

II. QOT ESTIMATION

The QoT estimation use case is treated as follows. First, the synthetic KB is described. Second, the feature selection is presented, followed by the construction and evaluation of the SVM and ANN classifiers considered for QoT estimation of unestablished lightpaths.

2.1 System Setup and data pre-processing

The KB used for QoT estimation was built using the data generation tool based on the Gaussian noise model described in [2]. The tool allows for channel BER estimation as a function of the linear and nonlinear noise contributions, as well as signal and link characteristics. Table I lists the signal and lightpath characteristics, assuming uncompensated coherent optical links as shown in Fig. 1. Additional system and link parameters (listed in Table II) are required to estimate the channel BER. For the purpose of this study, these parameters were assigned fixed values and were not fed to the ML models. The resulting synthetic KB includes 38,400 instances.

2.1.1 Feature engineering

The feasibility of using the Gaussian noise analytical model to estimate the QoT of a lightpath before establishment in a production network relies on the availability of all system and lightpath parameters listed in Tables I and II at the time of decision. In real-world scenarios, the link information (span length, span loss, link length, etc.) is often incomplete, inaccurate, or difficult to retrieve. In this context, ML could potentially be used to assist the QoT estimation process [9]. Furthermore, a synthetic KB includes bad QoT data which would be very difficult to get in the field.

First, an analysis of the impact of the different features in the QoT estimation process was performed with the aim to reflect the parameters availability in real-world applications. For that purpose, the classification accuracy of the SVM QoT estimator was assessed considering each feature separately. Six models were generated using 10-fold cross validation. The bigger the impact of the feature on QoT estimation, the higher the classification accuracy will be. The results are shown in Fig. 2, the most important parameter is the link length, followed by the number of spans, the modulation format, the span length, the channel power and the data rate.

Second, a Pearson correlation analysis performed on these 6 features showed a correlation coefficient of 0.33 between the span length and the link length, and 0.93 between the number of spans and the link length. For the purpose of this study, the

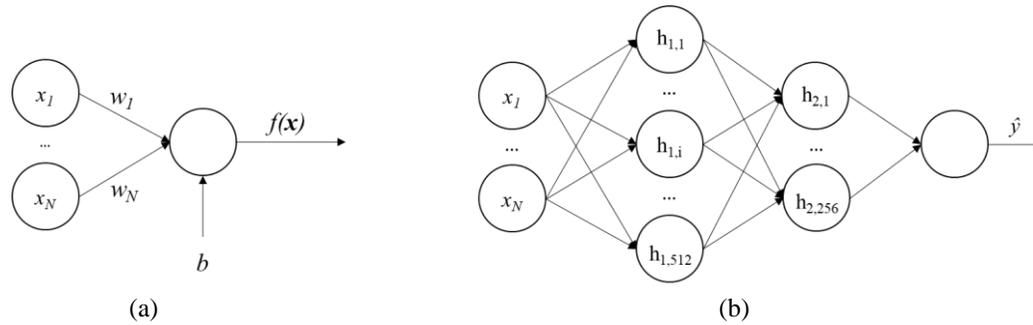


Fig. 3. (a) Neuron: the main component of the ANN; (b) Optimal architecture of the ANN model

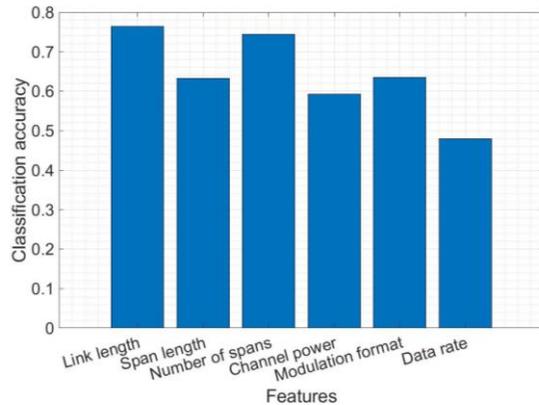


Fig. 2. Classification accuracy of the SVM QoT estimator obtained by considering one single feature

spans of a given link are assumed to be of equal length. Therefore, the resulting four parameters used to build the QoT estimation classifiers are the link length, the number of spans, the channel power and the modulation format. In real-world scenarios where the spans are of variable length (except in submarine cable links), the choice of the four parameters would likely be different as the lightpath SNR depends on the longest span in a link.

2.2 Construction of the QoT estimation models

2.2.1 SVM model

SVM is the first model considered for the binary classification of lightpaths into good or bad QoT in this study. This is a popular supervised learning method that constructs an optimal hyperplane as a decision surface with a maximized margin of separation between the two classes in the data. The optimal location of the decision surface is determined by using a small subset of the training observations, referred to as support vectors.

SVM models were built using the four features resulting from the feature engineering performed in subsection 2.1.1. The hyperparameters optimization process was performed using a 80/20 split ratio for training and test datasets, respectively, and a 5-fold cross validation, with respective ranges for C and γ of $[1 \times 10^{-5}, 1 \times 10^5]$ and $[1 \times 10^{-5}, 1 \times 10^3]$. These are typical ranges for the two hyperparameters that decide the performance of the SVM model. The resulting SVM classifier was trained with the 30,720 instances as in [8].

The optimum hyperparameters C and γ obtained with the Gaussian kernel, providing the best classification accuracy, are 1×10^3 and 0.5102, respectively. These values are comparable to the ones obtained with the complete set of features ($C = 1 \times 10^3$, $\gamma = 1.4$) [8]. The SVM classifier was built using MATLAB R2018a and the LIBSVM library for SVM in [11].

2.2.2 ANN model

The ANN is proposed as the second model for QoT estimation. Its main component is the neuron where it computes an output $f(x)$ value based on an input vector x , a set of weights w and a bias b as shown in Fig. 3(a). Combining multiple of these neurons on multiple layers increase the ANN capacity of model the data space and generally improve its estimation performance. To adjust the weights and biases of the ANN, it requires a learning phase through the observation of a training database. During that phase, the learning algorithm performs backpropagation to update the ANN parameters according to the estimation error [10].

The KB was split into a training set and a test set with a 80/20 ratio. Also, the training set is divided into a training set and a validation set. The latter serves as an evaluation set for the selection of hyperparameters such as the learning rate, the size of the layers and the number of layers. The ANN was trained with the Adam optimizer for mini batches of 512 samples. Leaky ReLU was selected as the activation function of the neurons except for the last layer where it was designed with the sigmoid function. These activation functions were chosen based on their effectiveness demonstrated in numerous studies. Furthermore, to account for the class imbalance in the KB, the weighted binary cross-entropy has been used as the loss function of the ANN.

Evaluation of the ANN hyperparameters determined their values to consider for the final model design. The number of hidden layers was set to two, the size of the hidden layers was set to 512 neurons for the first hidden layer and 256 neurons for the second hidden layer and the learning rate was set to 0.00074. This architecture was arbitrarily selected, whereas the number of hidden layers and their sizes were determined through a manual process using fixed ranges of values for both parameters. An automatic optimization process performed with the Tree-Structured Parzen Estimator and 50 different rate values allowed determining the optimal learning rate. For the number of hidden layers, up to four layers were evaluated with the first hidden layer set to 128 neurons and half the amount of the previous layer for subsequent hidden layers. For

Table III Confusion matrices of the SVM and ANN QoT estimators for different feature sets

			Predicted "0"	Predicted "1"
All 6 features (link length, span length, number of spans, modulation format, channel power, data rate)	ANN	True "0"	5,855	14
		True "1"	20	1,791
	SVM	True "0"	5,847	21
		True "1"	27	1,785
Top 4 features (link length, number of spans, modulation format, channel power)	ANN	True "0"	5,336	565
		True "1"	26	1,753
	SVM	True "0"	5,614	275
		True "1"	158	1,633
Top 3 features (link length, number of spans, modulation format)	ANN	True "0"	4,941	960
		True "1"	190	1,589
	SVM	True "0"	5,531	359
		True "1"	522	1,268

Table IV QoT estimation accuracy and computation time of the SVM and ANN models for different feature sets

	ANN		SVM	
	Accuracy	Computation time (ms)*	Accuracy	Computation time (ms)*
All 6 features (link length, span length, number of spans, modulation format, channel power, data rate)	99.56%	0.276	99.38%	3.45
Top 4 features (link length, number of spans, modulation format, channel power)	92.30%	0.214	93.30%	1.01
Top 3 features (link length, number of spans, modulation format)	85.03%	0.268	88.52%	0.935

* The models were executed on a system with an Intel® Core™ i5-8600K 3.6 GHz CPU, 16 GB RAM and a GTX 970 GPU.

the size of the hidden layers, 64, 128, 256, 512 and 1024 neurons for the first layer were tested with half of the amount of the previous layer for subsequent layers. The resulting ANN model is shown in Fig. 3(b). The size of the input vector is N and depends on the number of link features. The neural network output \hat{y} is the probability (ranging from 0 to 1) that the link has an acceptable QoT. Python 3 and the Pytorch package were used to build the ANN model.

2.3 Evaluation of the QoT estimation models

The estimation performance of the SVM and ANN models was evaluated with different sets of input features. The confusion matrices shown in Table III give a clear insight of true positive and true negative rates achieved with the SVM and ANN models over the test dataset for the different feature sets.

As the number of bad QoT (76.67% of the KB) is higher than the number of good QoT, the recall and F1-score are used as performance metrics in addition to accuracy. These metrics determine the correctly predicted classes in the case of unequal class distribution. The classification accuracy and the computation time in making class predictions for the different classifiers implemented with SVM and ANN are also calculated as performance metrics and presented in Table IV.

In this study, we did not focus on the training time, which can be very high, as we tried to measure how fast the proposed ML algorithms can estimate the QoT. No pre-processing has been performed on the data. Attributing higher costs to misclassifications was tested with the SVM as a class balancing method, but no significant improvement was observed in the classification results.

The QoT estimation accuracy of the ANN model with the complete feature set is 99.56% with a recall and a F1-score of 98.90% and 99.05% respectively. With the top four features, the accuracy decreased to 92.30%, the recall to 98.54% and the F1-score to 87.57%. Finally, with the top three features, the ANN obtained an accuracy of 85.03%, a recall of 89.32% and a F1-score of 73.43%. The SVM QoT estimator achieved classification accuracy of 99.38% and 98.50% and 98.66% for the F1-score and recall respectively with the complete feature set. For the top four features, the accuracy is 94.3% with 91.18% and 88.29% for the recall and F1-score. Finally, for the top three features, the accuracy is 88.52% with 70.84% and 74.22% for the recall and F1-score respectively.

The SVM QoT estimator shows better classification accuracy with reduced feature sets whereas the ANN classifier performs slightly better with the complete feature set. As for the computation time, the number of features does not affect

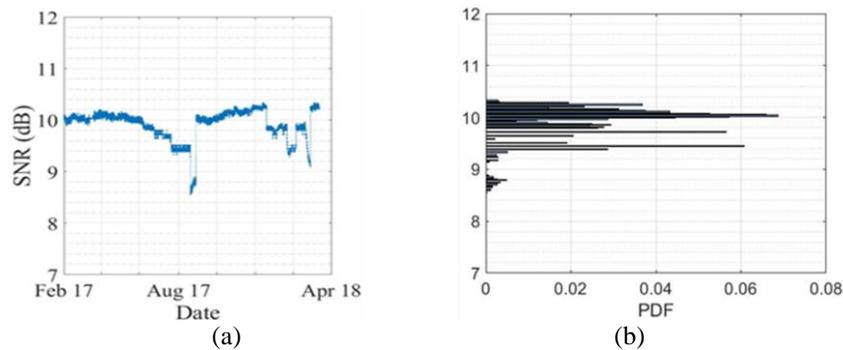


Fig. 4. (a) SNR vs. time; (b) PDF of SNR data

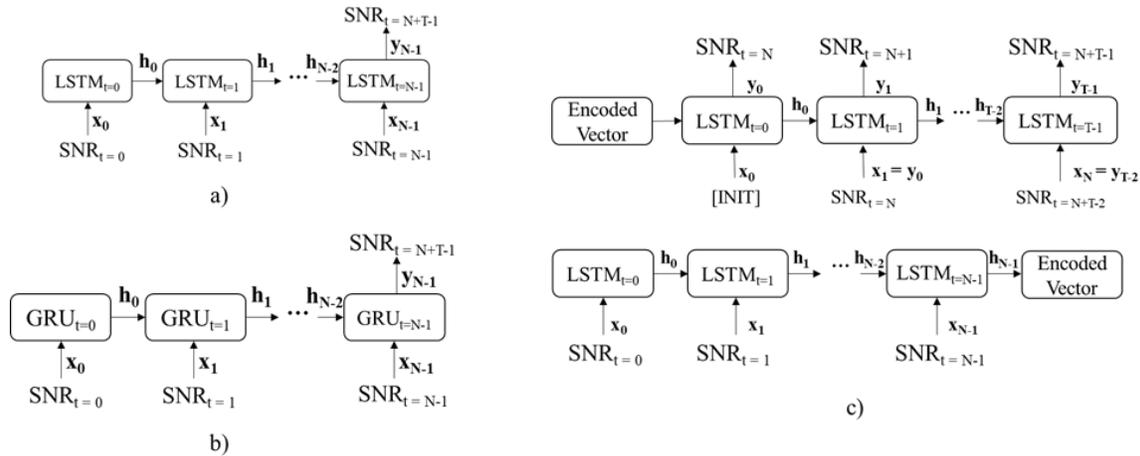


Fig. 5. RNN topologies of the SNR forecast models: a) LSTM b) GRU and c) Encoder-Decoder LSTM

the ANN models whereas it shows drastic change between the two reduced sets and the complete set.

III. SHORT-TERM SNR FORECAST

In this section, the SNR forecast use case is presented as follows. First, the KB used in the experiment is described. Second, the SNR prediction models and their design choices are detailed. Finally, the prediction results of these models are shown and discussed.

3.1 Construction of the SNR prediction models

The KB used for performance prediction was constituted of field performance monitoring (PM) data collected at a 15-min sampling rate over 13 months for a PM-QPSK channel at 100 Gb/s in a 1,500-km production link of the CANARIE network.

The SNR observations computed from the raw BER data are shown in Fig. 4. The probability density function (PDF) of the series is shown in Fig. 4(b). During the observation period, the BER varied between 6.2×10^{-3} and 5×10^{-4} with a mean value of 9.32×10^{-4} . It is interesting to note that the theoretical BER value calculated using the Gaussian noise model ($\sim 1 \times 10^{-4}$) is consistent with the observed BER values. An augmented Dickey-Fuller (ADF) test for unique root and a Kwiatkowski-Phillips-Schmidt-Shin (KPSS) test for trend stationarity were performed on the time series of PM data using built-in for ADF and KPSS tests, respectively). The peak-to-peak functions in MATLAB (0.05 significance level) revealing that the time series was not stationary (p-values of

0.62 and 0.01 amplitude of the 24-hour and 7-day seasonal components of the SNR time series (0.0275 dB and 0.0840 dB, respectively) was found to be quite small. The 1,020 missing data instances in the SNR time series (over 38,203 instances) were replaced by average values observed over a 30-day moving window.

3.1.1 LSTM model

The LSTM (Fig. 5(a)) is a type of recurrent neural network which can identify patterns in time series and use them to make predictions. LSTM models use structures called *gates* to control the cell states and a combination with the input information to determine the outputs. For each SNR value in the input sequence x of size N , the LSTM updates its internal states h_t until it reaches the last value in the sequence ($\text{SNR}_{t=N-1}$). At that time step, the LSTM produces an output y corresponding to the SNR forecast for a horizon of T hours ($\text{SNR}_{t=N+T-1}$).

The LSTM model, like the other models, was built with the aim to forecast the SNR change based on historical field data. For building the prediction models, the KB was split in the ratio 80/20, which corresponds to training and test datasets of 10 and 3 months, respectively. A validation set, containing 20% of the training set was used to determine the appropriate hyperparameters for each model. Thus, the other hyperparameters of the models, such as the learning rate, dropout rate, teacher forcing ratio and size of the hidden layer have been tested for different ranges ((0.00001, 0.000025 and 0.00005), (0 and 0.2), (0, 0.25 and 0.5) and (256, 512),

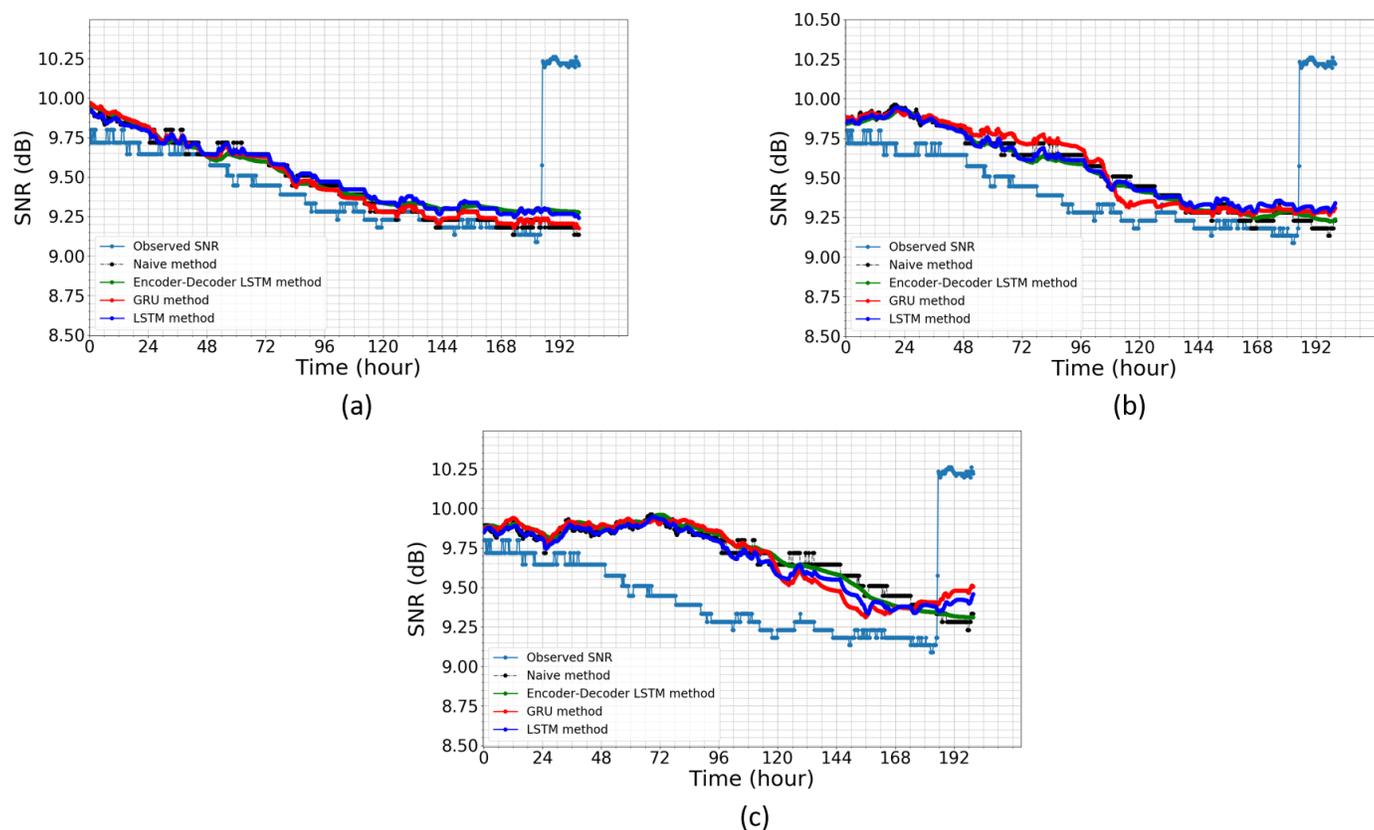


Fig. 6. Predicted vs. observed SNR: (a) 24-hour horizon; (b) 48-hour horizon; (c) 96-hour horizon (period of February 14th, 2018 to February 22nd, 2018 of the test set)

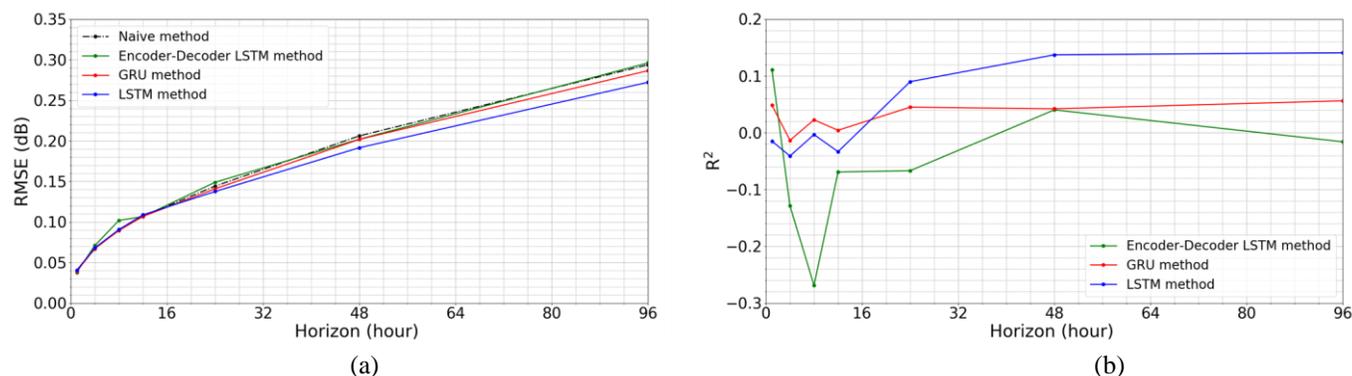


Fig. 7. (a) RMSE vs. forecast horizon on the test set; (b) R^2 vs. forecast horizon

respectively) using the Root Mean Square Error (RMSE) as the performance metric. The validation phase allowed setting the learning rate to 0.00001, the size of the hidden layer to 256 neurons, the dropout rate to 0.2 and the teacher forcing rate to 0. The Keras package in Python 3 was used for building the LSTM model.

3.1.2 Encoder-Decoder LSTM model

For sequence-to-sequence prediction task where the input is a sequence of multiple values and the model output is also a sequence, one can use an encoder-decoder LSTM topology. First, the encoder LSTM (Fig. 5(c)) processes all the entries in

the input sequence x and updates its internal states h_t at each input time step. When it reaches the last input value ($SNR_{t=N-1}$), it produces an encoded vector containing the state of the gates and cells. Then, the decoder LSTM starts predicting each value for the output sequence y based on the encoded vector and passes its internal states h_t to itself for the next prediction time step. The size of the prediction sequence y is T , the forecast horizon. Also, the input x of the decoding LSTM corresponds to the previous prediction except for the first predicting time step where it could be initialized to 0, a random value or the last value in the input sequence. For this work, the latter is selected.

To train the model, the KB was split in the ratio of 80/20 for the training and test sets with 20% of the training set being used for validation. The hyperparameters to be evaluated are observation window size, the learning rate, the size of the hidden layer, the dropout rate and the teacher forcing rate. At the validation phase, 0 was obtained for the last two hyperparameters, while the learning rate was set to 0.0005 and the size of the hidden layer to 512 neurons. Three learning rate values (0.00025, 0.0005 and 0.001) were evaluated. Learning rates typically range from 0 to 1. Generally, a low learning rate leads to longer convergence time to an acceptable validation score while a high learning rate might create instability during training and not converge. Thus, the goal was to select a value which produces a good validation score while not taking too long to converge.

3.1.3 GRU model

The GRU model works in the same way as the LSTM model (Fig. 5(b)). The difference resides in its internal states structure. It contains less gates for controlling the cell states thus it is less complex to implement and faster to run. The same training set and test set ratio was used for the GRU. Also, the LSTM hyperparameters evaluation process was applied to the GRU model. The dropout rate and the teacher forcing rate were set to 0 as in the validation process for the Encoder-Decoder LSTM model, while the learning rate and the size of the hidden layer were assigned to 0.00001 and 256 neurons, respectively. The Keras package in Python 3 was used for building the GRU model.

3.2 Evaluation of the SNR prediction models

As in [7], the size of the observation window (i.e. the size of the SNR sequence used by the model to determine the next SNR value for a given forecast horizon) has a significant impact on the model performance. Thus, the lowest RMSE, determined for the range of observation windows of 24, 48 and 96 hours, for each model was obtained with a 48-hour observation window during the validation phase except for the Encoder-Decoder LSTM where the lowest RMSE was obtained with a 96-hour observation.

Three metrics were used to evaluate the performance of the models: the RMSE, the R^2 and the computation time. The R^2 , in Fig. 7(b), was used to evaluate the model performance with respect to baseline model. The higher the value the better the model is to predict the future SNR values while a large negative score indicates that it is excessively worse than the naive method at predicting SNR values. A R^2 score of 0 means that the model and the naive method has identical predicting performance. Moreover, the performance of the SNR prediction models was assessed for forecast horizons ranging from 1 to 96 hours and compared to a naive method assuming that the SNR at a given time is equal to the SNR at latest value in the observation window. The RMSE obtained for the entire test set are shown in Fig. 7(a).

The RMSE increases as a function of the forecast horizon for all the models, as expected, with a maximum improvement of 0.022 dB at 96 hours for the LSTM over the naive method.

Fig. 6 shows the observed SNR for three 192-hour periods in the test dataset for the period of February 14th, 2018 to

February 22nd, 2018 as well as the predicted SNR for three different forecast horizons (24 hours, 48 hours and 96 hours). The objective was to determine if the SNR could be forecast with time scales of several hours.

For the LSTM model, the R^2 starts with -0.0148 at 1 hour and gradually improves with higher forecast horizon ending with a value of 0.1415 at 96 hours. With this network, it could outperform the baseline method for forecast of 24, 48 and 96 hours. The LSTM was the best model among the tested models for forecasting at higher horizons with the highest R^2 and lowest RMSE values. It could not beat the naive method at lower horizons (1 hour to 12 hours). Also, in Fig. 6(a) and Fig 6(b), we see that the LSTM behave similarly to the naive method for the 24-hour and 48-hour forecast horizon. For longer term forecast, in Fig. 6(c), the LSTM prediction curve is slightly less distant from the observation curve than the other three models. This could partially explain why the LSTM has the lowest RMSE among the studied techniques.

For the GRU model, the R^2 values are mostly positive ranging from -0.0133 (4-hour prediction) to 0.0563 (96-hour prediction) indicating that it gets slightly more accurate prediction than the naive method except for a 4-hour horizon. Furthermore, the near zero GRU R^2 scores show that it behaves similarly to the naive method by predicting a SNR value close to the most recent value in the observation window. As shown in Fig. 7(a), the GRU RMSE curve follows the naive RMSE curve closely. This behaviour can be seen in Fig. 6(a) and Fig. 6(b). However, in Fig. 6(c), the GRU prediction curve seems to adopt a different shape than the naive one after 120 hours (5 days), becoming slightly less distant from the observations. Despite these observations, the GRU prediction curves for the 24-hour and the 48-hour horizons seem to be the ones closest to the observed values curve. Overall, the GRU model was the best along all the horizons because it has the most positive R^2 values.

Finally, for the Encoder-Decoder LSTM, the R^2 values range from -0.2688 (8-hour prediction) to 0.1112 (1-hour prediction). The Fig 7(b) shows that the Encoder-Decoder LSTM could slightly outperform the naive method for the 1-hour and 48-hour forecast horizons. It mainly follows a similar behaviour as the naive method due to its R^2 score mostly close to 0. Among all the tested models, the Encoder-Decoder LSTM obtained the worst results. However, it got the best performance at very short-term forecast (1 hour). As expected, due to its R^2 mostly close to zero, we can see in Fig. 6 that, in all cases, the model follows the naive method prediction curve.

As for the R^2 score shown in Fig. 7(b), we can globally see that the GRU and LSTM have better performances compared to the naive method when predicting at higher horizons. The Encoder-Decoder LSTM does not show any R^2 trend as its performance with respect to the baseline model seems mostly random.

The models were executed on a system with an Intel® Core™ i5-8600K 3.6 GHz CPU, 16 GB RAM and a GTX 970 GPU. The computation time for the encoder-decoder LSTM increases with the forecast horizon. It ranges from 17.8 ms to 71.8 ms for forecasts of 1 to 96 hours. Meanwhile, the LSTM and GRU models computation time remained stable at around 47.7 ms and 40.8 ms respectively. The Encoder-Decoder

LSTM computation time varies with the forecast horizon T because it requires the decoder to predict T values while the LSTM and GRU models only predict one value.

IV. CONCLUSIONS

In this work, we substantially extended our previous study by introducing additional ML methods for QoT estimation and SNR forecast of lightpaths. A feature engineering process was applied to the synthetic BER KB to reflect the parameters availability in production networks. We believe that the results obtained with the SVM and ANN QoT estimators with SVM and ANN techniques shows the potential of ML for fast and automated lightpath provisioning. However, the performance of QoT estimators still remains to be validated with field data. Furthermore, the Recall and F1-score values for the classification of the unbalanced BER data obtained with reduced feature sets show that this approach would be applicable with real field data. The Encoder-Decoder LSTM and GRU models have been explored along with the LSTM model for SNR forecasting based on 13-month historical field data for one lightpath, with the objective to show the potential of ML in performance prediction of established lightpaths, through the identification of pattern and seasonality in the field SNR data. Satisfactory forecast accuracy was achieved in a supervised learning context, especially with the LSTM and GRU, using a single lightpath field KB.

Further work will focus on the optimization of the ML models with additional input data to search for complex patterns and periodicities in the SNR data that would allow forecasting the performance of established lightpaths with time scales of several hours, thus opening the way to proactive maintenance and network automation.

REFERENCES

- [1] Musumeci, F., Rottondi, C., Nag, A., et al.: "A Survey on Application of Machine Learning Techniques in Optical Networks", *IEEE Commun. Surveys & Tutorials*, Vol. 21, No. 2, pp. 1383-1408 (Second Quarter 2019).
- [2] Aladin, S., Tremblay, C.: "Cognitive Tool for Estimating the QoT of Lightpaths", *2018 Optical Fiber Commun. Conf. (OFC)*, San Diego, CA, 2018, paper M3A.3.
- [3] Sartzetakis, I., Christodouloupoulos, K., Varvarigos, E., "Accurate Quality of Transmission Estimation With Machine Learning", *IEEE/OSA J. Opt. Commun. Netw.*, 2019, Vol. 11, No. 3, pp. 140-150.
- [4] Choudhury, G., Lynch, D., Thakur, G., et al.: "Two Use Cases of Machine Learning for SDN-Enabled IP/Optical Networks: Traffic Matrix Prediction and Optical Path Performance Prediction", *IEEE/OSA J. Opt. Commun. Netw.*, 2018, Vol. 10, No. 10, pp. D52-D62.
- [5] Wang, Z., Zhang, M., Wang, D., et al.: "Failure prediction using machine learning and time series in optical network", *Opt. Express*, 2017, Vol. 25, No. 16, pp. 18553-18565.
- [6] Mo W., Gutterman C. L., Li Y., et al.: "Deep Neural Network Based Dynamic Resource Reallocation of BBU Pools in 5G C-RAN ROADMs Networks," *2018 Optical Fiber Commun. Conf. (OFC)*, San Diego, CA, paper Th1B.4.

- [7] Aladin S., Allogba S., Tran A. V. S., Tremblay C., "Recurrent Neural Networks for Short-Term Forecast of Lightpath Performance," *2020 Optical Fiber Commun. Conf. (OFC)*, San Diego, CA, paper W2A.24.
- [8] Tremblay C., Allogba S., Aladin S., "Quality of Transmission Estimation and Performance Prediction of Lightpaths Using Machine Learning [Invited]," *45th European Conf. Optical Commun. (ECOC)*, Dublin, Ireland, 2019, paper M.1.E.3.
- [9] Azzimonti, D., Rottondi, C., Tornatore, M., "Reducing Probes for Quality of Transmission Estimation in Optical Networks with Active Learning", *IEEE/OSA J. Opt. Commun. Netw.*, 2019, Vol. 12, No. 1, pp. A38-A48.
- [10] Rumelhart, D. E., Hinton G. E., Williams, R. J., "Learning representations by back-propagating errors", *Nature*, Vol. 323, pp. 533-536.
- [11] Chang C. C., Lin, C. J.: "LIBSVM: a library for support vector machines." *ACM Trans. Intelligent Syst. Technol.*, 2:27:1-27:27, 2011: <http://www.csie.ntu.edu.tw/~cjlin/libsvm>

Sandra Aladin received the B.Eng. degree in electronics from the Université d'Etat d'Haiti, Port au Prince, Haiti, in 2009 and the M.A.Sc. degree in telecommunications networks engineering from the École de technologie supérieure, Montréal, Canada, in 2018. Her Master's thesis focused on machine learning-based quality of transmission estimation in optical networks. Since 2018, she has been working as a research assistant on machine learning for optical networking with the Network Technology Lab, Department of Electrical Engineering, École de technologie supérieure, Montréal, QC, Canada.

Ahn Vu Stephan Tran received the B.Eng. degree in electrical engineering from École Polytechnique de Montréal, Canada, in 2013 and is completing a M.Eng. degree from the École de technologie supérieure, Montréal, Canada. His Master's project focused on machine learning-based quality of transmission estimation and performance prediction in optical networks. Since 2019, he has been involved in the application of machine learning algorithms to optical networks for lightpath classification and performance prediction.

Stéphanie Allogba received the B.Eng. degree in telecommunications from the ESME Sudria, France, in 2011 and the M.A.Sc. degree in telecommunications networks engineering from the École de technologie supérieure, Montréal, Canada, in 2015. Her Master's thesis focused on control and management algorithms for femtocell wireless networks. Since 2016, she is working towards the Ph.D. degree at École de technologie supérieure, with the Network Technology Lab. Her research interests include the application of machine learning algorithms for lightpath classification and performance prediction in optical networks.

Christine Tremblay received the B.Sc. degree in Engineering Physics from Université Laval, Quebec City,

Canada, in 1984, the M.Sc. degree (Energy) from INRS-Énergie, Varennes, Canada, in 1985 and the Ph.D. degree (Optoelectronics) from the École Polytechnique de Montréal, Canada, in 1992. She is a Full Professor with the Department of Electrical Engineering and Associate Director for the Ph.D. Program at the École de technologie supérieure. She is the Founding Researcher and Head of the Network Technology Lab. Before joining the ÉTS, she was a Research Scientist with the National Optics Institute (INO) where she conducted research on integrated optical devices for communication and sensing applications. She held senior R&D and technology management positions for several organizations. As Engineering Manager at EXFO and Director of Engineering at Roctest, she was responsible for the development of fiber-optic test equipment. She also served as Product Manager at Nortel for DWDM systems. Her team pioneered the research on filterless optical networking. Her current research interests include machine learning for optical networking applications, as well optical performance monitoring and advanced access networks for 5G applications. She has been co-instructor for SC314 and SC210 hands-on courses of the Optical Society of America on optical fiber and polarization measurements (2009-2015).

Dr. Tremblay is a member of the Optical Society of America (OSA), the IEEE Photonics Society, as well as STARaCom and COPL Strategic Clusters of FRQNT. She is currently serving as Program Chair for the Photonics Networks and Devices (NETWORKS) 2020 OSA Topical Meeting.