Multi-Step Span Loss Prediction in Optical Networks using Multi-Head Attention Transformers

Maryam Hedayatnejad, Yinqing Pei, David Boertjes, *Senior Member, Optica,* Dacian Demeter, Christian Desrosiers and Christine Tremblay, *Senior Member, IEEE*

***Abstract*—Span Loss is a pivotal characteristic of optical networks, and its accurate prediction enables adjustment for optimal performance and proactive monitoring. Deep learning models such as transformers, with their self-attention mechanism, have shown potential for various prediction tasks. In this study, we propose the Transformer-XL (Extra Long) model for single-step and multi-step forecasting, trained with field data. We report on models predicting span loss from 15 minutes to 5 days, using window sizes of 15 minutes to 10 days. The single-step model's average Absolute Maximum Error (AME) is better than the naive model by 2.13 dB and outperforms linear regression by 0.05-0.32 dB across different window sizes. Our single-step model also achieves better performance than the Recurrent Neural Network (RNN) with an AME improvement of 0.02 dB. The average AME of our multi-step model exceeds the naive model's performance by a range of 2.95-3.05 dB, linear regression by a substantial 0.02-0.15 dB and RNN by a range of 0.04-0.54 dB across different window sizes and forecast horizons. Based on Root Mean Square Error (RMSE), the single-step model performs better than the naive approach across various window sizes by 0.07 dB, achieves up to 0.07 dB improvement over linear regression, and delivers comparable results to RNN. Moreover, our multi-step model improves upon the naive approach with RMSE by 0.04 dB and RNN by 0.02 across various window sizes and forecast horizons. It also demonstrates a slight improvement over linear regression.**

***Index Terms*— Attention mechanism, deep learning models, multi-step prediction, optical networks, single-step prediction, span loss, time series forecasting, Transformer-XL, Transformers.**

I. Introduction

OPTICAL networks, serving as the backbone of modern communication infrastructures, have revolutionized the way data is transmitted across vast distances. These networks, being analog in nature, require continuous monitoring to ensure optimal performance. Key metrics, such as optical power levels, bit error rates, and signal-to-noise ratios, provide insights into the health and efficiency of these networks, acting as vital indicators of their operational status. Central to these metrics is the span loss, which is defined as the reduction in optical signal power caused by optical fiber attenuation and losses due to connectors and splices as it traverses through fiber optic cables. Monitoring span loss is imperative, as even slight increases can result in alterations in the signal-to-noise ratio, which directly affects the Quality of Transmission (QoT).

While real-time monitoring provides a snapshot of the current state, predicting future span loss can offer a proactive approach to network management. By anticipating potential issues, network operators can implement preventive measures such as cable repair or replacement, ensuring uninterrupted and high-quality data transmission. Machine learning, a subset of artificial intelligence that excels in recognizing patterns and making predictions based on data, offers a powerful solution to this problem. Within machine learning, deep learning models, characterized by their layered neural architectures, have shown utility in handling complex prediction tasks. Their ability to process vast amounts of data and extract intricate patterns makes them particularly suited for predicting span loss in optical networks.

Among the deep learning architectures, Transformer models have revolutionized tasks ranging from natural language processing to time-series forecasting. In the context of optical networks, the Transformer's ability to capture long-range dependencies and intricate relationships in data suggests promise for accurate and efficient span loss prediction.

In this study, we employ the Transformer with extra-long context model, better known as Transformer-XL. This model was pioneered by researchers at Google Brain and Carnegie Mellon University [1]. The "XL" suffix underscores its enhanced capacity to process extended contexts or sequences relative to its predecessors [1]. Our investigation encompasses both single-step and multi-step predictive models. We benchmarked these models against foundational models,

Manuscript received 7 April 2025; revised 8 May 2025.

This work was supported by the Mathematics of Information Technology and Complex Systems (Mitacs) of Canada under Grant IT16352 and by Ciena Corp. (*Corresponding author: Maryam Hedayatnejad.*)

Maryam Hedayatnejad and Christine Tremblay are with the Network Technology Lab, Department of Electrical Engineering, École de technologie supérieure, Montréal, QC, H3C 1K3, Canada (e-mail: maryam.hedayatnejad.1@ens.etsmtl.ca; christine.tremblay@etsmtl.ca).

Christian Desrosiers is with the Laboratory for Imagery Vision and Artificial Intelligence, Department of Software, and IT Engineering, École de technologie supérieure, Montréal, QC, H3C 1K3, Canada (e-mail: christian.desrosiers@etsmtl.ca).

Yinqing Pei is with Ciena, Ottawa, ON, K2K 0L1, Canada (e-mail: ypei@ciena.com).

David Boertjes is with Ciena, Ottawa, ON, K2K 0L1, Canada (e-mail: dboertje@ciena.com).

Dacian Demeter is with TELUS Corp., Edmonton, Canada (e-mail: dacian.demeter@telus.com)

https://.....

Digital Object Identifier ……

namely the naive and linear regression. The comparative analysis was conducted based on prediction error metrics Root Mean Square Error (RMSE), Absolute Maximum Error (AME), as well as the computational time required for each model's execution.

The rest of this paper is structured as follows: Section II presents related work. Section III defines span loss and explains how this parameter is calculated. Section IV describes the field data used in this study and focuses on data preprocessing, encompassing three facets: Extraction, Transformation, and Loading (ETL), addressing missing values, and statistical tests for time series stationarity. Section V outlines the principles of Transformer architecture and discusses its key advantages, including the use of multi-head attention for capturing long-range dependencies in data. Section VI describes the design of the Transformer forecasting models, as well as the parameter optimization process, and presents performance results for both single-step and multi-step prediction tasks. We conclude Section VII with a summary of our contributions and future directions of research.

II. Related Work

Optical Performance Monitoring (OPM) in optical networks ensures the reliability and efficiency of data transmission by providing real-time insights into the network's health. By predicting trends and changes in network parameters, operators can reduce downtime and maintain optimal performance, facilitate proactive management, and resolve issues before they arise. A broad range of methods based on machine learning have been proposed for performance prediction tasks. In [2], Tremblay et al. used 13-month field performance data to train Long Short-Term Memory (LSTM), Encoder-Decoder LSTM, and Gated Recurrent Unit (GRU) models for lightpath Signal-to-Noise Ratio (SNR) prediction over forecast horizons up to four days. In a related work, Mezni et al. demonstrated the robustness of a 1D Convolutional Neural Network over forecast horizons up to 24 hours for multi-step performance prediction using field bit error rate data [3]. Chouman et al. trained a Multilayer Perceptron (MLP) architecture using field data from 52 lightpaths deployed in two optical networks [4]. This study evaluated whether MLP and LSTM models could be used to forecast estimated received SNRs of established lightpaths compared to naive and linear regression methods.

In 2021, Allogba et al. proposed the use of two multivariate neural network models, based on GRU and LSTM methods, for predicting lightpath SNR over forecast horizons extending up to four days. These models were trained using field performance data and network features [5]. In a subsequent study [6], they reported superior performance of the single-step univariate LSTM model in comparison to the encoder-decoder LSTM and GRU models for the task of lightpath QoT forecasting. In 2023, Sun et al. proposed a method that uses receiver digital signal processing and nonlinear distortion analysis to accurately locate excess loss of wavelength selective switches (WSSs) and insufficient amplification of erbium-doped fiber amplifiers (EDFAs), without requiring additional monitoring hardware [7].

In the context of span loss analysis, Yaméogo et al. implemented a time-series decomposition method on a year's worth of span loss data from four bidirectional spans within a production network, with the objective of identifying long-term degradation trends in the fiber plant [8]. Following the extraction of the trend component, they applied the Mann-Kendall test to the span loss curves and utilized Sen's slope estimator test to determine the magnitude of span loss change, providing a deeper understanding of the fiber plant's degradation over time.

Despite its utility in trend analysis and basic forecasting, the time series decomposition method faces several challenges. It may struggle with nonlinear relationships and sudden changes in the data due to external factors such as maintenance activities or environmental impacts, which can mislead trend analysis and skew results. Moreover, statistical tests like Mann-Kendall and Sen's slope, though robust to non-normal distributions, do not adapt well to complex or abrupt changes in data structure [9]. Advanced deep learning models may offer an advantage in overcoming these limitations. In particular, transformers model complex dependencies in time series data without assuming linearity. Abrupt changes can be handled through their attention mechanisms, and learning from vast amounts of data to uncover subtle patterns that traditional methods might overlook [10]. Furthermore, the ability to dynamically weigh the importance of data points across time series allows for more precise and accurate predictions.

III. Span Loss: Definition and Calculation

Determining QoT in optical networks requires an understanding of the noise sources for each of the optical signals. This noise can be broken into two broad categories: linear and nonlinear. Linear impairments include filtering and noise from amplifiers called amplified spontaneous emission (ASE) [11]. Nonlinear impairments arise from the light interacting with the nonlinear medium of the fiber through the Kerr effect. Nonlinear impairments can be modeled as a noise source using the GN model [12]. The magnitude of these noise sources is influenced by the span loss wherein the power launched at the head-end of the fiber drives the non-linear noise while the power at the tail-end of the span determines the linear signal-to-noise ratio. There is an optimal launch power which balances these two noise sources for a given amplifier setup, fiber type and span loss [13].

A. Span Loss Definition

Span loss is defined as the total attenuation of optical power as the optical signal propagates through a designated segment or span. Contributors to this loss include factors such as fiber attenuation, splice and connector loss, as well as fiber bending [8]. Service Providers (SPs) extract and analyze span loss data to perform root cause analysis (RCA) on problems such as component degradation, and cable aging [8]. Careful monitoring and management of span loss is imperative for the preservation of long-term stability and optimal performance of optical networks.

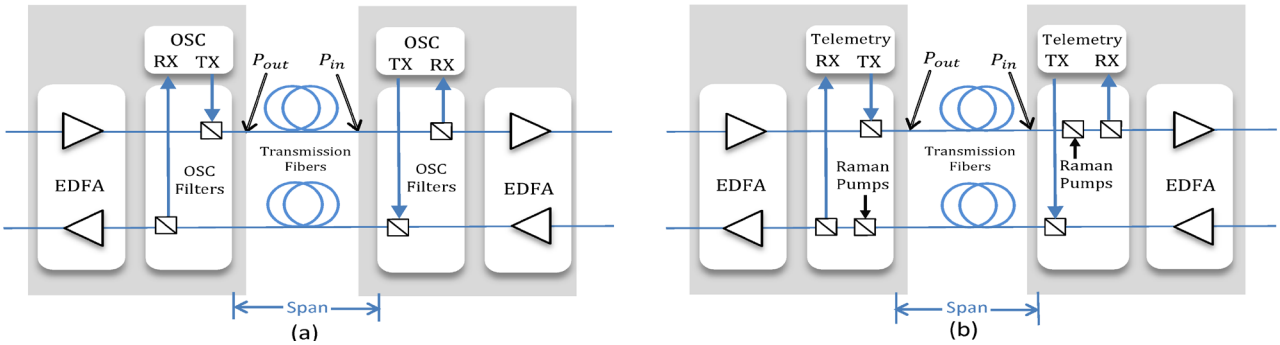

Fig. 1. Optical line system configuration: (a) without Raman amplification; (b) with Raman amplification.

B. Span Loss Calculation

Fig. 1 shows the optical line system configuration with and without Raman amplification. The span loss is defined as the sum of all losses in the fiber segments between sites including the attenuation of the optical fiber as well as the loss of the patch panels, splices, and connectors.

In this study, the calculation of span loss was performed automatically by the Network Elements (NEs) using (1) or (2), depending on whether Raman amplification is used, respectively:

$$Span\ Loss = P_{in} - P_{out} \tag{1}$$

$$Span\ Loss = P_{in} - P_{out} + G_{Raman} \tag{2}$$

where P_{in} and P_{out} indicate input power and output power of a probe channel, respectively, and G_{Raman} is the Raman gain.

In optical links without Raman amplification (Fig. 1(a)), the probe used for input and output powers is the optical supervisory channel (OSC) channel at 1510 nm which is referenced through in-factory calibration to the faceplate connectors connecting to the transmission fibers.

In optical links with Raman amplification (Fig. 1(b)), the probe used for input and output powers is the telemetry channel at 1527 nm again referenced to the faceplate connectors. The Raman gain is measured at turn up using the telemetry channel. The wavelength of this channel is chosen to be close to the traffic carrying channels but just outside the gain range of the Erbium-Doped Fiber Amplifiers (EDFAs) such that it is limited to the span of interest and experiences a similar Raman gain as the channels themselves.

IV. Field Data and Data Preprocessing

Performance Monitoring (PM) data were collected at 15-minute sampling intervals over the course of 18 months for 532 fiber spans in a production network in Canada. Data collection was performed in optically amplified links of lengths ranging from 6 to 138 km. The resulting 532 time series of span loss data were divided into two categories, stable or dynamic. Fiber spans were considered dynamic if the span loss changed by at least 2 dB between the highest and lowest values recorded during the observation period. Most spans were considered stable, while only 58 fiber spans (11%) were labeled dynamic. In the dataset, each span is a bidirectional pair of fibers. Among the 58 dynamic spans, 30 spans exhibited dynamic characteristics in both fiber directions and 28 spans showed dynamic loss variation in only one direction.

The dataset comprised 58 distinct time series, representing dynamic spans with various features, including Card ID, Slot number, Port number, Shelf number, Date and Time, and Span loss values. For training the ML models in this study, we utilized the Date and Time and Span loss value features. The Date and Time feature provided temporal context, while the span loss value was the primary metric for forecasting future span loss amounts. Each of the time series had 51,936 samples of span loss. The data preprocessing pipeline involves essential steps, outlined as follows:

A. Extracted, Transformed, and Loaded (ETL)

In the initial phase of data preprocessing, the measured span loss data was extracted, transformed, and loaded (ETL) from the database. The network topology was recovered from network provisioning tools and stored in a proprietary Manage, Control and Plan (MCP)-export file, containing detailed information such as the Terminal Identifier (TID), the Access Identifier (AID) comprising the shelf, slot, and port as well as the source, destination, and path information for each fiber in each span.

B. Missing Values in the Time Series

Among 58 time series, 51 had missing values totaling 13% or slightly higher relative to their total data. The remaining 7 time series had missing values, ranging between 27% to 33%. We used the strategy of reindexing followed by interpolation with the nearest value for filling the gaps in the time series. Reindexing involved adjusting the data for a consistent time index, filling gaps where data points were missing, and ensuring a complete sequence for subsequent interpolation.

C. Statistical Tests for Time Series Stationarity

To gain insights into the underlying structure of the dataset, we first conducted an exploratory analysis based on time series decomposition and autocorrelation. Results indicated that most of the series exhibited a stationary trend. To further substantiate this finding, we employed the Augmented Dickey-Fuller (ADF) test and the Kwiatkowski-Phillips-Schmidt-Shin (KPSS) test,

two well-known statistical methods for evaluating time series stationarity. The analysis yielded results indicative of 67.24% trend-stationarity (39 time series out of 58), 24.13% difference-stationarity (14 time series out of 58) and 8.62% stationarity (5 time series out of 58). Consequently, a first-order differencing of time series was implemented as a preprocessing step. Subsequent reapplication of the ADF and KPSS tests to the transformed time series affirmed the stationarity of the series, except for two instances. To mitigate the influence of outliers within the dataset, the transformed samples were subsequently subjected to normalization utilizing a robust scaler.

V. Transformers and Their Multi-Head Attention Mechanism

Transformers were introduced by Vaswani et al. in 2017 for natural language processing (NLP) [10]. They have demonstrated superior performance in modeling sequential problems, surpassing Recurrent Neural Networks (RNNs) [14]. Compared to RNNs, transformers leverage self-attention mechanisms to process data in parallel rather than sequentially. This approach captures long-range dependencies, enhancing the model's pattern recognition capabilities.

Central to the success of transformers is the attention mechanism. By assigning attention scores to various parts of the input, this mechanism dictates the model's focus during prediction, allowing simultaneous consideration of different elements. This not only overcomes the limitations of previous sequential models in capturing long-range dependencies but also offers interpretability through attention scores. The reduction in computational time, owing to parallel processing, adds to its advantages [15]. The self-attention mechanism in transformers can be described as follows [10].

1) **Calculate Query, Key, and Value matrices**
Query, Key, and Value matrices are derived from the input and are used to calculate the attention scores:

$$Q = X.W^Q, K = X.W^K, V = X.W^V \quad (3)$$

where X is the input, and W^Q, W^K and W^V are weight matrices.

2) **Calculate Attention Scores**
The attention scores are computed using the dot product of the Query and Key matrices, followed by scaling, and applying the SoftMax function:

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_K}}\right)V \quad (4)$$

where Q, K and V represent the query, key, and value matrices, respectively, and d_K is the dimensionality of the key vectors.

3) **Multi-head Attention**
Transformers often use multiple attention heads to capture different aspects of the relationships in the data. The outputs of these heads are concatenated and linearly transformed as represented in (5).

$$\text{MultiHead}(Q; K; V) = \text{Concat}(\text{head}_i)W^O$$
$$\text{where head}_i = \text{attention}(QW_i^Q, KW_i^K, VW_i^V) \quad (5)$$

In this study, the Transformer-XL model was utilized for single-step and multi-step span loss forecasting based on historical data. In contrast to the Vanilla Transformer model, which uses a fixed-length context in language modeling and consists of identical layers with self-attention mechanisms [10], the Transformer-XL model uses a recurrence mechanism across segments. The model can therefore handle a larger context than the vanilla Transformer's fixed-length constraint, which allows it to remember and use information from earlier portions of the data sequence more efficiently and improves its ability to understand and process long data sequences [16]. Moreover, unlike Vanilla Transformer's absolute positional encoding, Transformer-XL introduces relative positional encoding, enhancing the model's awareness of token positions within the sequence. As a result, the model can also capture dependencies between segments and handle long sequences more efficiently by reusing computations from previous segments [1].

VI. Architecture and Performance Analysis of the Forecasting Models

This section is organized into two main subsections. The first subsection details the implementation and parameter optimization for both single-step and multi-step Transformer-XL model forecasting. The second subsection presents the performance results and discussion, structured into separate subsections for single-step and multi-step models.

A. Architecture and Parameter Optimization

For single-step forecasting, we used the Transformer-XL model to predict span loss one time step ahead, corresponding to a 15-minute interval. To explore the impact of historical context on prediction accuracy, we used measured span loss sequences of various lengths—25, 50, 100, and 200 hours— as input into the model. On the other hand, for multi-step forecasting, as with single-step models, the measured span loss values, characterized by varying sequence lengths, are used as input for the model. Depending on the input sequence length, the model is designed to predict span loss at different forecast horizons. Specifically, for input lengths of 48, 144, and 240 hours, the model predicts span loss for forecast horizons of 24, 48, 96, 72 and 120 hours. For clarity, models are designated based on their input sequence lengths, measured in hours. For example, a model that processes an input length of 48 hours, is labeled as "Transformer-XL 48". Consequently, we considered three distinct transformer models, corresponding to our varied input lengths of 48, 144, and 240 hours.

The single-step and multi-step Transformer-XL models were implemented in Python 3 using the PyTorch package. For model training, we divided the dataset into training and test sets using a ratio of 80:20. A total of 51,937 samples are included in

each dataset, of which 10,388 are used for testing and the others for training purposes. The development and optimization of the model involved two distinct categories of hyperparameters. The first category relates to the training process and includes the input sequence length (represents the amount of historical data), the output sequence length (represents the forecast horizon, which is set to one for single-step forecasting), the learning rate, the batch size, and the number of epochs. Model learning dynamics and convergence are controlled by these hyperparameters, as well as how quickly and accurately the model adapts to underlying patterns. The second category of hyperparameters defines the model's architecture, such as the dimensionality of the model (d-model), the number of layers (num-layers), dropout rate (dropout) and the number of attention heads (nhead). These structural hyperparameters control the complexity and capacity of the model.

TABLE I
OPTIMIZATION OF MODEL HYPERPARAMETERS

Hyperparameter	Tested Values	Optimum Value
Learning rate	0.001,0.0001, 0.00001, 0.000001	0.00001
d-model	64, 128, 256, 512	128
Number of layers	2, 3, 4, 5, 6, 7, 8	6
Number of heads	2, 4, 6, 8	4
Optimizer	Adam, AdamW, Adamax, SGD, ASGD, Adamgrad, RMSprop, Adadelta	Adamax
Dropout rate	0, 0.1, 0.2	0
Batch-size	22, 25, 32, 40, 50	32
Number of epochs	20, 30, 50, 100	20

The different values considered for each hyperparameter are reported in Table I. While several values were tested for d-model, including 128, 256, and 512, we found that the models were not highly sensitive to this hyperparameter and that a value of 128 worked well in most cases. The model's performance is also affected by the window size, which corresponds to the number of observations N in the sequence of inputs (in single-step forecasting the prediction horizon is always set to 15 minutes).

B. Model Performance Results and Discussion

Following training, the performance of the models was evaluated on an independent test set spanning roughly 3.6 months selected temporally from the last portion of the data. We compare the model's results with linear regression, naive and RNN models, giving a detailed view of its effectiveness in forecasting tasks. This approach aligns with prior studies [4], [5], and [17] that used these approaches as comparison baselines.

The naive model is a simple and straightforward forecasting approach that predicts the future value of a time series as the last observed value in the observation window. Specifically, for span loss forecasting at horizon T, the naive model assumes that the span loss value will be equal to the most recent value in the observation window. This can be represented as:

$$y_{t+1} = y_t \tag{6}$$

where y_{t+1} is the predicted span loss value at time t+1 and y_t is the observed value at time t. Despite its simplicity, this method often achieves better results than more complex models [4, 5] , which are more sensitive to noisy time series.

On the other hand, the linear regression model learns a linear relationship between the span loss value at horizon T and past values within the observation window. It fits a linear equation to the observed data, minimizing the sum of squared differences between the observed and predicted values. The model then uses this relationship to predict future values. The linear regression equation can be expressed as:

$$y_T = \beta_0 + \beta_1 y_{t-1} + \beta_2 y_{t-2} + \cdots + \beta_n y_{t-n} \tag{7}$$

where y_T is the predicted span loss at horizon T, y_{t-1}, y_{t-2}, …, y_{t-n} are the past span loss values, and β_0, β_1, …, β_n are the coefficients learned by the model. The linear regression model employed in this study was derived from the neural network module of PyTorch.

RNN model handles sequential data by maintaining an internal state, allowing it to capture temporal dependencies. RNNs are effective for time series forecasting and sequence prediction. We implemented the RNN using the same data pipeline as the Transformer model and both models were implemented using the PyTorch library.

Two performance metrics were employed in the evaluation phase of the models' performance as observed in previous works [5, 6]: 1) the RMSE, which measures the average magnitude of squared errors between predicted and observed values and then computes its root; 2) the AME, which measures the largest singular error made by the model, thus highlighting its potential worst-case performance. Finally, we also computed the training time of the model, which offers insights into its efficiency and speed.

1) **Single-step model performance results**
For single-step models, the evaluation was conducted over a variety of window sizes of 15 minutes, 25 hours, 50 hours, 100 hours, and 200 hours, which corresponds to input sequence lengths of 1, 100, 200, 400, and 800, respectively. We selected an initial input sequence length of 1 to align with the naive model that inherently uses a single input to forecast subsequent values. Despite the simplicity of this structure, the transformer still outperformed the naive, linear regression and RNN models, demonstrating its efficacy in making accurate predictions even with minimal input data. To compare the performance of all models, we computed the RMSE and AME for each time series. Then, to get a comprehensive measure, we averaged the RMSE and AME values across all time-series. Thus, when we refer to RMSE and AME of the models in this study, we refer to their averages across all analyzed time series. Fig. 2 shows the distribution of RMSE values obtained by the

Transformer-XL model for 58 time series across different window sizes. The RMSE values of some time series are higher than the average, which is mainly due to the presence of anomalies in these time series that lead to an increase in prediction errors. These anomalies may result from data irregularities or sudden fluctuations in the observed values, which can be caused by various factors such as monitoring device malfunctions, network disruptions, or unexpected external events. The results of this study are summarized in Fig. 3, which compares the performance of our single-step Transformer-XL model against baseline models. As shown, the proposed model achieves RMSE/AME values of 0.0924/3.454 for a window size of 15 minutes, 0.0884/3.453 for a 25-hour window, 0.0891/3.452 for a 50-hour window, 0.0894/3.453 for a 100-hour window, and 0.0920/3.450 for a 200-hour window. The single-step Transformer-XL consistently outperformed the naive baseline, achieving a 0.069-0.073 dB lower RMSE and a reduction in AME of at least 2.13 dB, across various window sizes. Compared to linear regression, it achieved an RMSE improvement of 0.014-0.071 dB and an AME improvement of 0.047–0.315 dB, while surpassing the RNN with RMSE improvements of 0.002-0.01 dB and AME improvements of up to 0.02 dB. However, as window sizes increase, the difference in RMSE between the Transformer and RNN models decreases, likely due to added noise in larger windows, which reduces the Transformer's performance advantage. In terms of AME, while the values for linear regression, RNN, and Transformer-XL were relatively close, the Transformer-XL consistently outperformed both models, showcasing superior worst-case performance. Notably, its AME values remained steady across all window sizes, further underscoring the model's robustness in varying conditions.

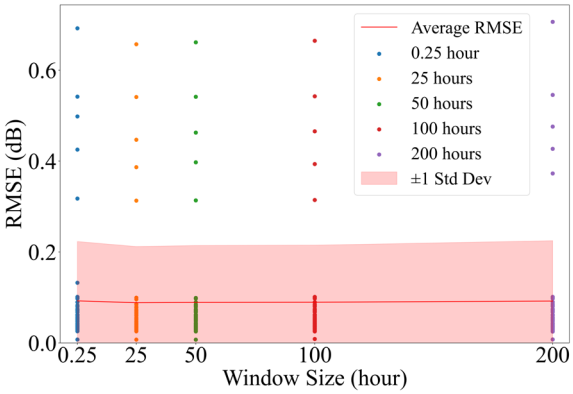

Fig. 2. Variation of RMSE for 58 time series across different window sizes.

Computation time is a critical metric for assessing model efficiency. We measured the training time of each time series across different window sizes. We observed an increase in computational time as the input length increases, from 5.5 minutes with a window size of 1 (15 minutes) to 1.8 hours for a window size of 800 (200 hours). Transformers are particularly sensitive to input size, as their self-attention mechanism requires computing a separate score for each input pair, leading to complexity that grows quadratically with the number of inputs. This increase in computational time confirms the scalability challenges of transformers in relation to input size and underscores the inherent trade-off between input size and computational efficiency.

2) **Multi-step model performance results**

Following training, the multi-step models were evaluated using the same test dataset and same metrics (RMSE, AME and computation time) as the single-step models. RMSE was calculated in two ways: based on the last predicted value and all predicted values. In this study, given the importance of evaluating how the model predicts across all forecast horizon time steps, we chose to focus on the RMSE calculated from all predicted values. This approach offers insights into the model's consistency and accuracy at each step of the forecast, rather than relying solely on its ability to predict the final value accurately. Fig. 4 compares the performance in RMSE and AME of the multi-step Transformer-XL model with naive, linear regression and RNN baselines. The Transformer-XL-48 model obtained RMSE/AME values of 0.0989/3.461 dB and 0.0995/3.468 dB for 24-hour and 48-hour forecast horizons, respectively. In comparison, the Transformer-XL-144 model recorded RMSE/AME values of 0.0988/3.462 dB, 0.0998/3.464 dB, 0.0996/3.467 dB, and 0.1008/3.545 dB for 24-hour, 48-hour, 72-hour and 96 hour forecast horizons, respectively. Lastly, the Transformer-XL-240 exhibited

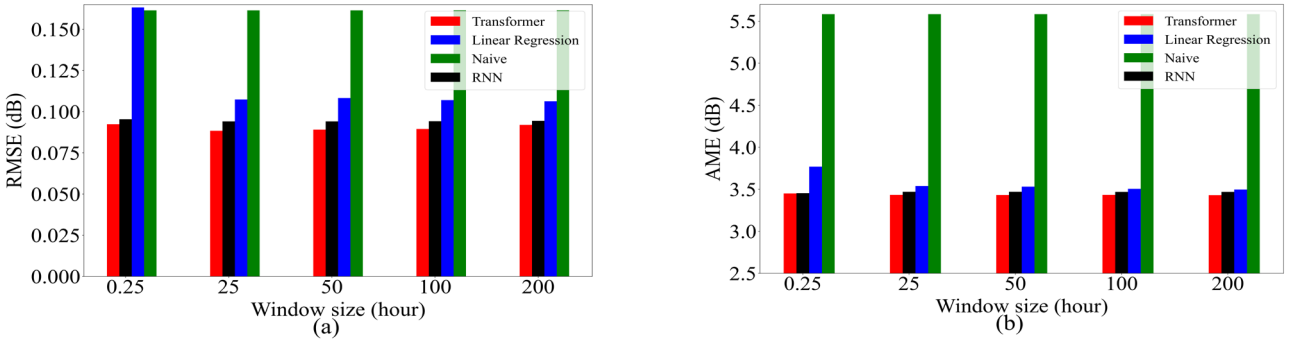

Fig. 3. Performance comparison of single-step Transformer-XL with baseline (naive, linear regression, RNN) models: (a) RMSE; (b) AME.

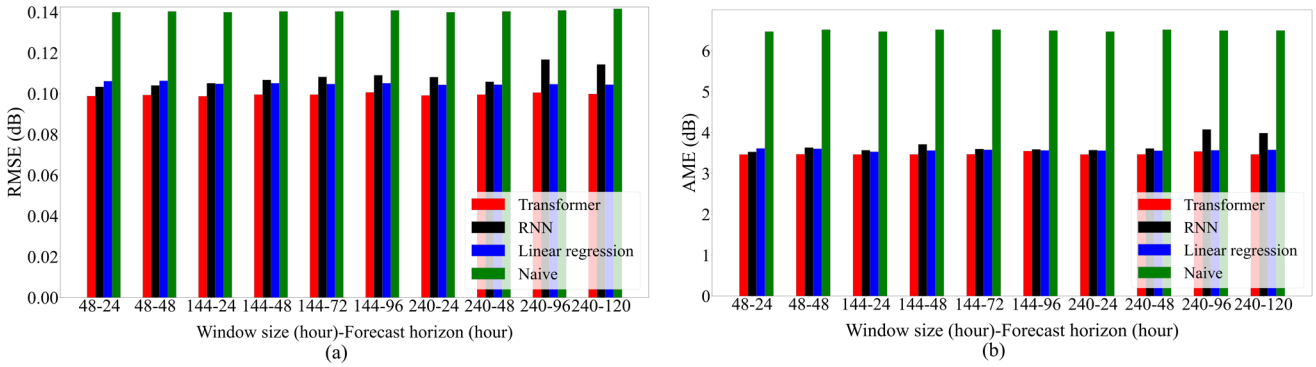

Fig. 4. Performance comparison of multi-step Transformer-XL with baseline (naive, linear regression, RNN) models: (a) RMSE; (b) AME.

RMSE/AME values of 0.0991/3.462 dB, 0.0998/3.465 dB, 0.1008/3.536 dB, and 0.1000/3.465 dB, respectively, for forecast horizons of 24 hours, 48 hours, 96 hours, and 120 hours. As reported, the best performance was achieved for the 24-hour horizon. Results also show that the highest RMSE and AME values were observed for the 144-hour and 240-hour window sizes across the 96-hour forecast horizon. Upon closer examination, the similarity of RMSE values across the 48-hour and 96-hour forecast horizons for both 144-hour and 240-hour window sizes, along with the closely aligned AME values, indicates the model's ability to capture long-term dependencies. This aligns with the expected performance of Transformer-XL in handling longer data sequences without compromising precision.

As illustrated in Fig. 4, the Transformer-XL model consistently outperformed naive model across all forecast horizons, with RMSE differences ranging from 0.0406 dB to 0.0419 dB and AME differences ranging from 2.948 to 3.051 dB. Furthermore, the results show that the performance gap between the transformer and naive models widened as the forecast horizon increased. Similarly, comparing Transformer-XL with linear regression models reveals that Transformer-XL consistently surpassed them in terms of RMSE and AME values across all combinations of window sizes and forecast horizons. The improvements vary, with RMSE gains ranging from 0.004 to 0.01 dB and AME gains from 0.017 dB to 0.149 dB. Transformer-XL also outperformed RNN models, demonstrating lower RMSE and AME across all window sizes and forecast horizons. It achieved RMSE improvements of 0.02 dB and AME gains ranging from 0.04 to 0.54 dB. Looking deeper into the results, RMSE for a 24-hour forecast horizon revealed some interesting trends: 0.0989 dB for the 48-hour window size, 0.0988 dB for the 144-hour window size, and 0.0991 dB for the 240-hour window size. The 144-hour window provides the best balance between historical context and predictive accuracy, achieving the lowest RMSE and making it the optimal configuration for accurate short-term forecasting in this study. The slight increase in RMSE for the 240-hour window size suggests that more historical data does not always lead to better accuracy, as it adds unnecessary complexity. Additionally, similar RMSE values may result from several factors. First, the transformer's self-attention mechanism effectively weighs different parts of the input data, capturing relevant patterns even in longer sequences. Second, its model capacity, with multiple heads and layers, allows it to handle both short and long sequences with comparable accuracy. However, longer sequences can introduce more computational overhead, especially during training. This comparison between single-step and multi-step computation time shows that despite similar accuracy levels, resource requirements may be higher, such as increased processing time or greater memory usage.

In addition to the performance results, it is important to acknowledge the differences in model complexities. For single-step prediction, the computational complexity of the naive model at inference is O (1), as it simply predicts the last measured value. In comparison, linear regression has a complexity of O (n), with n being the number of past observations (window size), whereas each layer of the Transformer-XL has a complexity of O ($n^2 \times d \times h$), where d is the feature dimension of the layer and h is the number of attention heads. The quadratic term arises from the self-attention mechanism, which requires computing an attention weight between each pair of tokens (corresponding to past observations) in the network. Although this mechanism enables Transformer-XL to capture intricate patterns and dependencies in time series data and crucial for accurate forecasting, it also increases computational resources, training time, and inference time. The trade-off between complexity and accuracy is a critical consideration in model selection, depending on the specific requirements and constraints of the application. In real-world production networks, even small improvements in prediction accuracy can have a significant impact on proactive monitoring, justifying the use of more complex models like Transformer-XL.

VII. Conclusion

In this study, we explored the use of transformers in forecasting tasks, specifically focusing on the Transformer-XL models trained with field data for predicting the span loss in production networks. For the single-step forecasting task, the models were designed to predict span loss ranging from an hour

to five days, utilizing window sizes ranging from 25 hours to 10 days. In single-step forecasting, our model outperformed the naive method, showing an average RMSE improvement of 0.07 dB across diverse window sizes. It also shows a range of 0.014-0.071 dB improvement over linear regression and a range of 0.002-0.01 dB over RNN. Notably, in terms of average AME, the Transformer-XL model significantly outperformed the naive approach by about 2.13 dB and linear regression by 0.047-0.315 dB, though less significantly compared to the RNN, up to 0.02 dB across all window sizes. For multi-step forecasting, in terms of RMSE, the Transformer-XL model outperformed the naive model by 0.0406-0.0419 dB across varied window sizes and achieved a 0.004-0.007 dB and a 0.004-0.016 dB improvement over linear regression and RNN, respectively. For average AME, the Transformer-XL model beated the naive model by 2.948-3.051 dB, linear regression by 0.017-0.149 dB and RNN by 0.0420-0.5380 across all window sizes.

This paper has showcased the advantages of employing the Transformer-XL model for predicting span loss. Future research will focus on enhancing the model's interpretability by analyzing how the multi-attention mechanism identifies key patterns, providing deeper insights into its decision-making process. Additionally, validating these findings with larger PM datasets featuring diverse attributes and network topologies would strengthen the model's applicability. Moreover, the type of span (with or without Raman amplification) was not tracked in the dataset used in this study; how Raman amplification may affect the prediction accuracy would deserve further investigation in future works. The lower RMSE and AME values across various forecast horizons indicate that the Transformer-XL models consistently outperform basic naive, linear regression, and RNN models, particularly as the forecast horizon increases. This suggests that the model is well-suited for further exploration with larger datasets and longer prediction intervals, potentially in real-world applications across different domains.

REFERENCES

[1] Z. Dai, Z. Yang, Y. Yang, J. Carbonell, Q. V. Le, and R. Salakhutdinov, "Transformer-XL: Attentive language models beyond a fixed-length context," *ACL 2019 - 57th Annu. Meet. Assoc. Comput. Linguist. Proc. Conf.*, pp. 2978–2988, 2020, doi: 10.18653/v1/p19-1285.

[2] C. Tremblay, S. Allogba, and S. Aladin, "Quality of transmission estimation and performance prediction of lightpaths using machine learning," *IET Conf. Publ.*, vol. 2019, no. CP765, pp. 5–7, 2019, doi: 10.1049/cp.2019.0757.

[3] A. Mezni, D. W. Charlton, C. tremblay, C. Desrosiers, "Deep Learning for Multi-Step Performance Prediction in Operational Optical Networks," *Conf. Lasers Electro-Optics*, pp. 33–34, 2020.

[4] H. Chouman, P. Djukic, C. Tremblay, and C. Desrosiers, "Forecasting Lightpath QoT with Deep Neural Networks," *2021 Opt. Fiber Commun. Conf. Exhib. OFC 2021 - Proc.*, pp. 4–6, 2021, doi: 10.1364/ofc.2021.th4j.5.

[5] S. Allogba, S. Aladin, and C. Tremblay, "Multivariate machine learning models for short-term forecast of lightpath performance," *J. Light. Technol.*, vol. 39, no. 22, pp. 7146–7158, 2021, doi: 10.1109/JLT.2021.3110513.

[6] S. Allogba, S. Aladin, C. . Tremblay, S. Sandra, and C. . Tremblay, "Machine-Learning-Based Lightpath QoT Estimation and Forecasting," *J. Light. Technol.*, vol. 40, no. 10, pp. 3115–3127, 2022, doi: 10.1109/JLT.2022.3160379.

[7] K. Sun, Z. Yu, H. Huang, X. Dong, and K. Xu, "Intra-Node Excess Power Loss Localization Based on Signal Nonlinear Distortion in Optical Fiber Links," *49th European Conference on Optical Communications (ECOC 2023)*, *Hybrid Conference, Glasgow, UK*, vol. 2023, no. 34, pp. 1055–1058, 2023, doi: 10.1049/icp.2023.2433.

[8] B. L. M. Yameogo, D. W. Charlton, D. Doucet, C. Desrosiers, M. O'Sullivan, and C. Tremblay, "Trends in Optical Span Loss Detected Using the Time Series Decomposition Method," *J. Light. Technol.*, vol. 38, no. 18, pp. 5026–5035, 2020, doi: 10.1109/JLT.2020.3000967.

[9] M. Collaud Coen *et al.*, "Effects of the prewhitening method, the time granularity, and the time segmentation on the Mann-Kendall trend detection and the associated Sen's slope," *Atmos. Meas. Tech.*, vol. 13, no. 12, pp. 6945–6964, 2020, doi: 10.5194/amt-13-6945-2020.

[10] A. Vaswani *et al.*, "Attention is all you need," *Adv. Neural Inf. Process. Syst.*, vol. 2017-Decem, no. Nips, pp. 5999–6009, 2017.

[11] E. Desurvire, "The golden age of optical fiber amplifiers," *Biochem. (Lond).*, vol. 30, no. 6, pp. 8–10, 2008, doi: 10.1042/bio03006008.

[12] P. Poggiolini, G. Bosco, A. Carena, V. Curri, Y. Jiang, and F. Forghieri, "The GN-model of fiber non-linear propagation and its applications," *J. Light. Technol.*, vol. 32, no. 4, pp. 694–721, 2014, doi: 10.1109/JLT.2013.2295208.

[13] M. Lonardi *et al.*, "Optical Nonlinearity Monitoring and Launch Power Optimization by Artificial Neural Networks," *J. Light. Technol.*, vol. 38, no. 9, pp. 2637–2645, 2020, doi: 10.1109/JLT.2020.2985779.

[14] M. Dehghani, S. Gouws, O. Vinyals, J. Uszkoreit, and Ł. Kaiser, "Universal transformers," *7th Int. Conf. Learn. Represent. ICLR 2019*, pp. 1–23, 2019.

[15] A. Katharopoulos, A. Vyas, and N. Pappas, "Auto Regression Transformer," *Proc. 37th Int. Conf. Mach. Learn.*, pp. 5156–5165, 2020, [Online]. Available:http://proceedings.mlr.press/v119/katharopoulos20a.html.

[16] Y. Liao, K. Meng, J. Zhang, and G. Liu, "Unleashing the Potential of Attention Model for News Headline Generation," *Proc. Int. Jt. Conf. Neural Networks*, 2020, doi: 10.1109/IJCNN48605.2020.9207206.

[17] S. Aladin *et al.*, "Recurrent neural networks for short-term forecast of lightpath performance," *Opt. InfoBase Conf. Pap.*, vol. Part F174-, no. 34, pp. 2020–2022, 2020, doi: 10.1364/OFC-2020-W2A.24.

Maryam Hedayatnejad diploma in Mathematics from the National Organization for Development of Exceptional Talents (NODET) and B.Eng. in Electrical Engineering in 2004 in Iran. She began her professional journey as a warranty manager in an injection molding machine manufacturer. In 2011, she joined Telecommunication Infrastructure Company (TIC) in Iran. Then, completed M.Sc. in Electrical Engineering in 2014 in Iran. Her master's thesis focused on leveraging a multi-agent approach for telecommunication network traffic control using game theory. She continued her professional endeavors until 2020 in TIC. From 2020 till now, she is pursuing her Ph.D. at École de technologie supérieure, within the Network Technology Lab. Her research primarily revolves around employing machine learning algorithms for anomaly detection and performance prediction in optical networks.

Yinqing Pei received the B.Eng. degree and PhD. from Beijing University of Post and Telecommunication, Beijing, China in 2008 and 2015. She joined Ciena as an engineer in 2013. Her

research interest is characterization, control, and optimization of optical line systems with both analytical and machine learning methods.

David W. Boertjes BSc physics, University of New Brunswick, Fredericton NB, Canada 1993. MSc physics, Dalhousie University, Halifax NS, Canada 1995. PhD electrical engineering, University of Alberta, Edmonton AB, Canada 1998. From 1995 to 1998 David researched active and passive polymer optics as low-cost photonic components for use in optical telecommunications at TRLabs in Edmonton AB. Canada. In 1998 he joined Nortel's Optical Networks division. In 2010 he was part of the Nortel MEN team acquired by Ciena. Currently he is a senior director in R&D working on physics-based software applications and network automation with Ciena based in Ottawa Canada. He has also served on and acted as chair of two OFC subcommittees. Dr. Boertjes is currently a senior member of Optica and received Ciena's Distinguished Engineer Award in 2014.

Dacian Demeter "To be completed."

Christian Desrosiers (Ph.D. in Applied Mathematics, Polytechnique de Montréal, 2008) is a Professor in the Software Engineering and IT at ÉTS Montréal, Quebec University. He is the co-director of the LIVIA Lab on Imaging, Vision and AI, and holder of Industrial Research Chair in Computer Vision for Industrial Applications. Professor Desrosiers has published over 150 papers in machine learning, computer vision, and medical imaging.

Christine Tremblay (Senior Member, IEEE) received the B.Sc. degree in engineering physics from Université Laval, Quebec City, Canada, in 1984, the M.Sc. degree in energy from INRS-Énergie, Varennes, Canada, in 1985, and the Ph.D. degree in 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 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 the Product Manager at Nortel for DWDM systems. Her team pioneered the research on filter less optical networking. Her current research interests include machine learning for optical networking applications, as well as optical performance monitoring, fiber sensing and quantum optical networking. She is co-instructor for SC528 short course at OFC 2024 as well as for SC314 and SC210 hands-on courses from 2009 to 2015. Dr. Tremblay is a member of the Optical Society of America (Optica) and STARaCom and COPL Strategic Clusters of FRQNT. She served as Program Committee Member for the OFC 2020-2023 Subcommittee N3: Architecture and software-defined control for metro and core networks. She also serves as Member of the OFCnet Committee for OFC since 2024.