

Pro-active Performance Monitoring in Optical Networks using Frequency Aware Seq2Seq Model

Rishabh Jain, Umesh Sajjanar



Abstract: Performance Monitoring (PM) and Fault Detection have always been a reactionary approach in Optical Networks for most service providers. Any kind of fault (power surge, ageing issues, equipment faults and failures, natural calamities, etc.) in an optical network is detected only after the fault has occurred and mitigation is performed afterward. The resultant service outages for end-users cause huge financial and reputation losses to the vendors. Therefore, there is a strong need for proactive detection of faults to limit disruption and provide uninterrupted services to clients. We achieve this objective by doing a multi-horizon time series prediction of Bit Error Rate at the receiver end of an optical circuit using our custom designed Frequency aware Sequence to Sequence (FaS2S) Neural Network. The predicted value of BER can be used to notify users of failure scenarios before they occur. Further corrective action, such as automatic re-routing or manual intervention can then be taken by the user. With this model, we can even configure the network properties dynamically during periods of low BER to push the network efficiency to its maximum capacity. See [inference Video](#) for BER inference capabilities of FaS2S

Keywords: Performance Monitoring, Optical Networks, Artificial Intelligence

I. INTRODUCTION

Multiple factors can affect the quality of transmission in an optical circuit. The most critical parameter while measuring such changes is the Bit Error Rate (BER) which is very sensitive to even minute changes/disturbances in the environment. Circuits become optically infeasible after BER crosses a certain threshold. Even other important parameters such as Modulation format/Baud Rate/No. of circuits provisioned on a fiber must be decided based on the current signal quality and hence it is crucial to actively monitor BER. Most of the

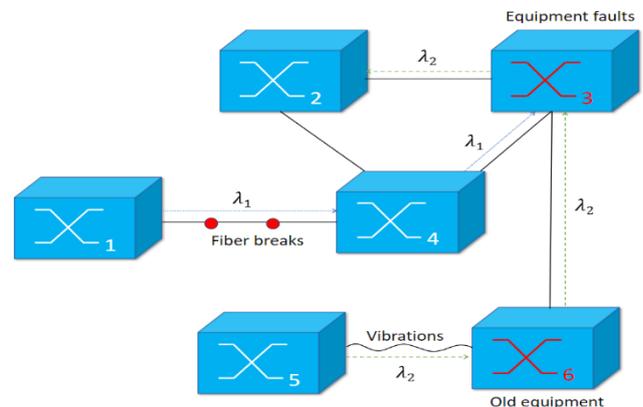


Figure 1. A sample mesh optical network having multiple BER impacting issues which can be temporary or permanent

service providers either prefer to operate their networks in a very conservative BER regime or employ a threshold-based protection [1]. In this scheme, another path is blocked/reserved after BER enters a warning region so that quick switching can be done in case of a failure. Both approaches have several drawbacks compared to proactive monitoring. The first limitation is the inefficiency of the Optical Networks in terms of bandwidth due to sub-optimal settings. There is also a big overhead of blocking another wavelength in a protection-based scheme even if the BER may just briefly cross the warning region and come back. This would also result in new service requests getting rejected due to congestion. In the case of low latency decay (very fast deterioration), an early prediction will provide more time to take appropriate action whereas large timescale deterioration (3-4 days) [2] forecasting can help in scheduling maintenance in advance. These advantages are not present in a protection-based approach. Various factors influence BER directly or indirectly. While fluctuations in module temperature cause a noticeable change in BER, power surges may lead to complete equipment failure. Any manufacturing issue or damage to the equipment caused by internal or external events results in an observable impact on the BER. In fact, according to Federal Communication Commission (FCC), more than one-third of service disruptions are caused by fiber-cable problems such as fiber breaks [3]. Natural calamities such as earthquakes & tsunami can also have a big impact on the optical cables. This impact is even more pronounced for the very long deep-sea optical networks. Researchers have even shown that you can use the change in performance monitoring parameters such as phase to detect the magnitude of earthquakes [4,5].

Manuscript received on 07 December 2022 | Revised Manuscript received on 26 December 2022 | Manuscript Accepted on 15 February 2023 | Manuscript published on 28 February 2023.

*Correspondence Author(s)

Rishabh Jain[†], Research Associate, Media and Data Science Research Lab, Adobe, Noida, India. Email: f2015550p@alumni.bits-pilani.ac.in, ORCID ID: <https://orcid.org/0000-0003-4305-6410>

[†]Work done while working at Cisco AIR Lab

Umesh Sajjanar^{*}, Engineering Manager, Cisco Artificial Intelligence Research Lab, Cisco Systems Inc, ORR Bangalore 560103 India. Email: usajjana@cisco.com, ORCID ID: <https://orcid.org/0000-0001-8823-5979>

© The Authors. Published by Lattice Science Publication (LSP). This is an open access article under the CC-BY-NC-ND license (<http://creativecommons.org/licenses/by-nc-nd/4.0/>)

There are a plethora of other real world factors such as moisture, humidity, vibrations, etc which can have an indirect effect on the performance of an optical circuit and are almost impossible to model using a knowledge based computation approach.

As these issues are very tough to model using a statistical/heuristic-based method due to the intrinsic complexity present in the data, machine learning forms an ideal candidate for pro-active monitoring. The advent of deep learning is redefining the way optical networks operate and they are already having a significant impact on how these networks function with applications ranging to fault detection, resource management, traffic prediction and lightpath establishment [6] [7] [8]. Inspired by these advances, we decided to tackle our problem by doing a multi-horizon time series prediction of BER by teaching a neural network to learn the signature of various events that affect BER from the data itself. Formally, the model is designed to recursively predict the probability for the time series regression problem:

$$p(b_{t+k,i}, \dots, b_{t+1,i} | b_{t,i}, x^{(h)}, \theta) \quad (1)$$

where $b_{t+1:t+k,i}$ is the Pre FEC BER prediction for the next k time steps at the i^{th} step of the series, $b_{t,i}$ is the historical data for Pre FEC BER, $x^{(h)}_{t,i}$ are the temporal covariates (variables which have a correlation with BER) present in the historical data and θ are the parameters of our neural network FaS2S.

Our contributions in this paper can be summarized as follows:

- We propose a novel end-to-end trainable **Frequency aware Sequence to Sequence model (FaS2S)** that explicitly utilizes higher frequency components from the time-series Performance Monitoring data.
- We propose a data pipeline to enable real time PM data collection, feature engineering, neural network training and inference for multi horizon forecasting of pre FEC BER.
- We achieve state of the art results for the task of optical circuit PM which in turn will provide important actionable intelligence to the network operator.

II. RELATED WORK

Several methods have been proposed for performance monitoring in optical circuits using BER or OSNR (Optical Signal to Noise Ratio) parameters. A lot of them are concerned with the estimation of the quality of lightpaths either before the circuit has been provisioned [9,10] or they perform an in-band estimation of circuits in real time [11,12]. Both approaches can not predict the outages in the future that are going to occur due to various unforeseen events and are only used for diagnosis. Some of them also do not have the capability to handle multi-variate inputs. Other approaches which do anomaly detection/diagnosis/alarm analysis [13,14] in optical networks have not considered high frequency components present in the PM data explicitly. Hence, their pipelines are not very sensitive to quick changes in the environment

which is a critical requirement for the fault detection application.

In [13], Liu et al. proposed an auto-encoder based approach for anomaly detection in optical networks with imbalanced data. Due to the scarcity of anomalous data (<3%) in their dataset, they trained their neural network to reconstruct the original signal and identify abnormality using the reconstruction loss. The problem with such an approach is that the range of normal data is so large that it is almost impossible to show the network every normal configuration during training. This would result in high false positives during deployment and cause unnecessary overhead for the operator. In [14], Zhang et al. used an RNN based method to perform alarm analysis and fault localization. They used negative data augmentation to tackle the unbalanced data problem. However, as they were still using an off-the shelf LSTM based model for predicting failures, their network was not responsive to sudden changes in the environment which resulted in less accuracy. As their work was doing alarm analysis, it is primarily useful for identifying the location of the fault but not for taking a mitigation step beforehand. In our work, we opted to explicitly use frequency information in our network to make it more sensitive to BER spikes present in the data. Moreover, apart from failure prediction ahead of time, FaS2S can also be used to change the network configuration dynamically.

FaS2S does a multivariate time series prediction by modeling the complex correlations of numerous factors affecting BER and hence forecasting its future patterns accurately. For our objective, we analyzed various state of the art methodologies used in time series prediction such as statistical based approaches, recurrent neural networks, attention-based models and Sequence to Sequence models but each of them have their own shortcomings. Statistical based methods [15] (exponential averaging, ARIMA) do not have the necessary degrees of freedom required to handle the complex patterns present in our data. Moreover, these methods are biased towards the most recent values and cannot capture the correlation between multiple input variables extracted from the optical network elements. Simple recurrent neural networks such as LSTM, GRU [16,17] have proven to be adequate for a single time step prediction and effectively handle the problem of vanishing gradient. However, their performance is not satisfactory for multi horizon forecasting [18] and more sophisticated methods have been established. Attention/Transformer [19,20] based models are suitable for NLP (Natural Language Processing) tasks due to the alignment advantages inherently present in them. For example, during neural machine translation, you want to pay attention to the context of a specific word while generating its opposite word in a different language. This kind of natural alignment is not present in our PM data and hence the attention mechanism does not offer any significant advantage to us while predicting the future sequence.

Furthermore, the attention layer looks at the whole historical sequence together and hence the neural network memory consumption becomes a lot higher due to many additional parameters. The training and inference time is also increased significantly which cause delay in the time bound decision making for critical faults. Seq2Seq models [21] have shown good results in forecasting the future values for generic sequences but fail to capture the high frequency components present in the signal, especially in our collected dataset of Performance Monitoring for Optical Network elements.

Researchers have recently started providing high frequency information present in the input signal explicitly or implicitly to the network. This has been shown to enable a better regularization of the neural network during training. For example, in Zflow[22], Chopra et al. implicitly conditioned the network to preserve contours while generating images for virtual try-on[23]. Virtual try on problem aims to synthesize an image of a person wearing an in-shop garment synthetically using Computer Vision techniques. After image generation the authors obtained an edge map of the human model by applying the Sobel operator [24] on the image. Subsequently, they imposed a smooth L1 loss against the edge map of the ground truth and trained the model end to end. Edges constitute the high frequency component in an image and by forcing the network to preserve their map, the network showed increased capabilities of faithfully reproducing the shape of the garment.

In Wavefill [25], the authors explicitly decomposed the image into different frequency bands using discrete wavelet transform for the image inpainting task[26]. Image inpainting involves filling missing pixels in an image such that the completed image is realistic looking and follows a similar probability distribution to the ground truth image. The authors achieved this objective by applying Haar transform on an image and separately predicting the different frequency component branches. To fuse these branches while avoiding inter frequency conflicts, they also proposed a Frequency region attentive normalization

scheme. Wavefill was able to achieve superior image inpainting performance on multiple datasets both qualitatively and quantitatively. Other methods have also used wavelets for different varying tasks such as super-resolution and image demoreing[27,28]. However, most of these approaches tackle image generation problem in the spatial domain which is inherently different from the prediction in time domain and hence require a different architecture. Using the above methods from computer vision as motivation, we design and implement a new architecture - Frequency aware Sequence to Sequence model for achieving state of the art results in multi horizon BER forecasting task.

III. DATA PIPELINE DESIGN

We design an appropriate data pipeline to smoothen out the process of inference so that new incoming data can be easily used to fine tune the model at regular intervals. This is especially important in the networking domain as new equipment is continuously added by service providers as part of upgradation of existing networks (Brownfield Scenarios). Each of the equipment/cable in the path of the circuit has its own loss function which can sometimes result in data that significantly deviates from the statistical distribution of older training data. Hence, it is imperative to train the network on recent data so that it can learn the current patterns and generalize well during deployment. We monitor and collect data from the receiver of an Optical Network. This is the place where optical to electrical conversion happens and data is read out for the final use case. Note that the PM data at the receiver is a function of the whole circuit path and will be affected by any disruption caused during the light traversal. In our case, we collate the data from CFP2 DCO (Digital Coherent Optical) plugin by Acacia. Collected data is cleaned, parsed and correlation analysis is done on it to identify the useful parameters/features of the data. Afterwards, scaling and normalization is done on the data to form the final features used by our neural network as shown in Fig. 2.

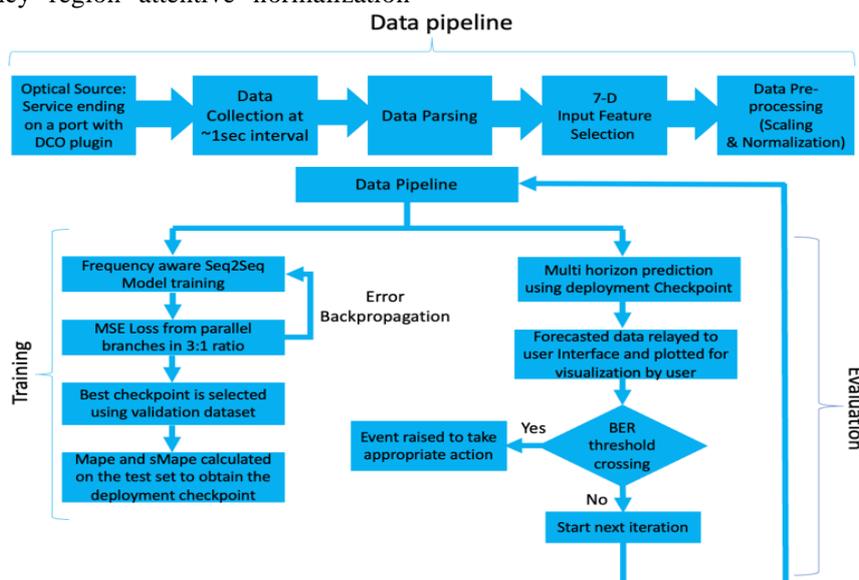


Figure 2. Data pipeline used by FaS2S for training and inference

The details of individual components of the pipeline are captured in the subsequent sections.

Data Collection As mentioned, optical signal quality is determined from the BER that is picked up from the coherent module present in the transponder/muxponder/ROADM cards. To test our model for real world scenarios and its generalization capabilities, we monitor and collect data from the receiver of multiple optical circuits in a lab environment. The topology of the circuits varies from a point-to-point connection to an optical circuit spanning up to 5 sites & consisting of multiple amplifiers and filters in between. BER is an accurate measure to estimate how the quality of the optical signal is changing. A low BER corresponds to a better signal. For our application, the BER is fetched from the coherent 400GE optical module, the CFP2 DCO that is attached to a transponder card. The CFP2 DCO Optical Interface is based on the Digital Coherent Optical transceiver (DCO) included in a CFP2 module form factor compliant to the CFP MSA HW specification 1.0. The DCO transceiver from Acacia supports a Symbol rate of up to 64GBd and high order QAM operations (up to 16 QAM), enabling the application of 400Gbps/wavelength. CFP2 DCO acts as the optical interface for the DWDM (Dense Wavelength Division Multiplexing) signal flowing into the network. Based on the signal received on the module, several inputs are collected that specify the performance of the module and the signal.

Data streaming and Feature Engineering The DCO plugin has the capability of reading multiple performance monitoring parameters which include:- 1) Laser transmitter Output Power, 2) Polarization Dependent Loss, 3) Differential Group Delay, 4) Second Order Polarization Mode Dispersion, 5) Received Input Power, 6) Module Temperature, 7) Received Q-Factor, 8) Chromatic Dispersion, 9) Received Coherent Channel Power, 10) Received Central Frequency Offset and 11) PRE Forward Error Correction(FEC) BER. We streamed the PM data at a sampling rate of ~1 sec to build our desired dataset in the

form of pandas data frame. Any streaming platform such as Kafka can be used for this purpose. We then performed feature engineering to separate out the relevant parameters useful for the prediction of BER. This step is important as you only want to pass the features which have a correlation with BER. Apart from some extra memory and computation overhead, passing useless features results in the neural network learning irrelevant patterns which hamper its ability to generalize at run time.

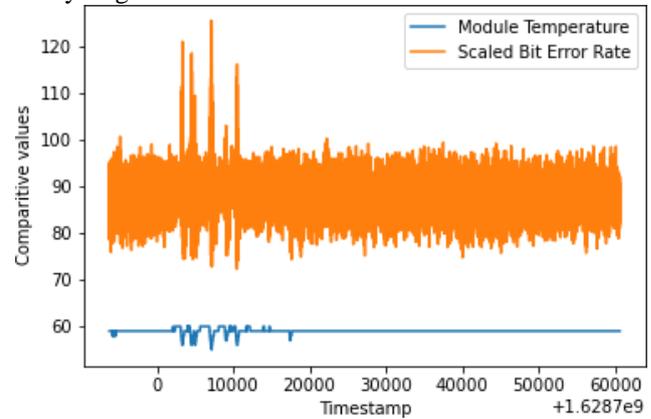


Figure 3. Feature validation showing non-linear relationship between Temperature and BER in data collected over 18 hours

To identify the key features, we made use of a combination of correlation matrix, optical expert's recommendation and manual validations. We first selected features that have a high correlation with BER and then looked at nonlinear relationships among the data. For example, even though the mathematical correlation operator between module temperature and BER shown in Fig 3 is comparatively less (-0.12), you can see that there is indeed a nonlinear correlation present between these variables. The spikes in BER occur when there is an instability in module temperature. The final 7 features selected after our analysis along with their physical significance are listed in Table 1.

Table 1. Final 7 input features selected using correlation matrix, expert recommendation and manual validation. These variables are used by our neural network for multi horizon Pre FEC BER forecasting for the next 5 minutes in steps of 15 seconds each

PM parameter	Physical interpretation
Laser transmitter Output Power	Total power transmitted by transceiver in dB
Received Input Power	Total power received by the receiver in dB
Module Temperature	Temperature at the DCO plugin in °C
Received Q-Factor	Q factor combines the SNR value for "0" bit and "1" bit to measure signal quality
Received Coherent Channel Power	Power measured for the particular wavelength at which the circuit is provisioned
Received Central Frequency Offset	Deviation of the actual wavelength from the tuned wavelength(-3200 to 3200 MHz).
PRE Forward Error Correction(FEC) BER	Historical Pre FEC BER data for the circuit being monitored

Scaling and Normalization Following selection, these parameters were averaged over a uniform period of 15 seconds each for the reduction of the random thermal noise inherently present in PM measurements. As the mean of the random noise is 0, this helps to smoothen out the curve for better predictions. The noise reduction is approximately $1/\sqrt{N}$,

where N is the sampling rate, assuming that the noise follows a Gaussian distribution. After averaging, we used min-max normalization to form the final features to be fed into the neural network. The final dataset consisted of 223,218 samples utilized for training, validation, and testing.

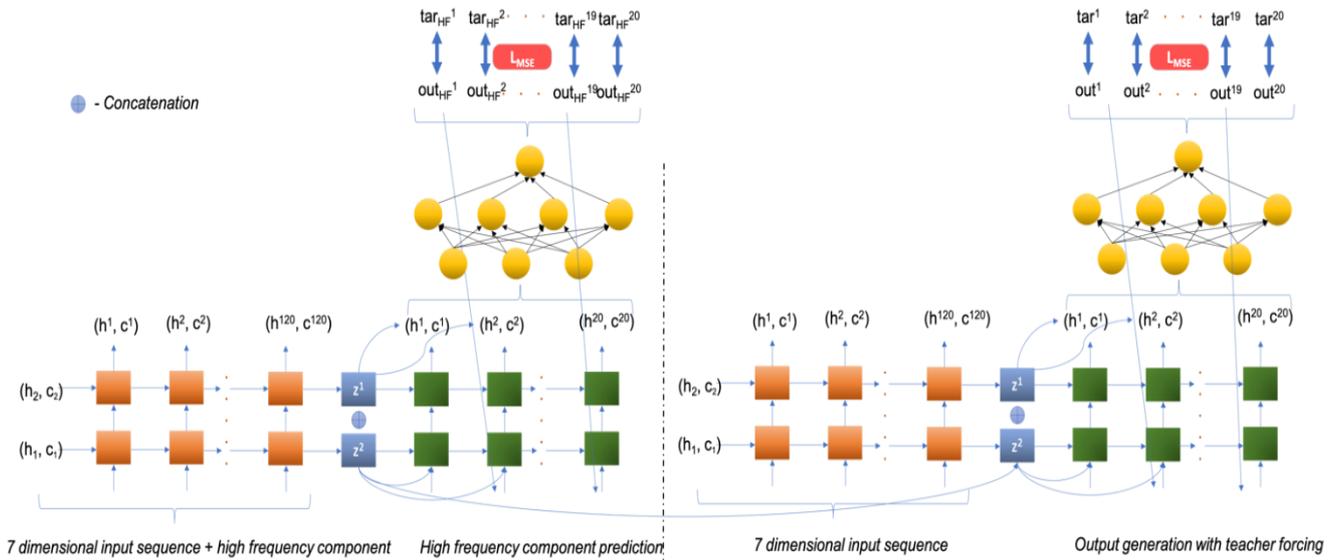


Figure 4. FaS2S consists of 2 branches, a high frequency (HF) prediction branch and the main future BER prediction branch. The context vector generated from the HF branch is explicitly passed to the main decoder to incorporate HF information in our main model

IV. ARCHITECTURE

We considered various algorithms for our use case that have been shown to produce reliable results for time series prediction. After analyzing various algorithms present in the literature, we decided to implement a custom-designed variant of Seq2Seq Neural network architecture. We define 120 timesteps (30 minutes) as the lookback period and we predict BER for the next 20 timesteps (5 minutes) in our model. It should also be noted that our architecture is scale invariant and can be easily modified to predict BER values for different timescales. Our network can mainly be broken up into 2 branches: A)

High Frequency (HF) prediction branch and a B) **Main prediction branch**. Both the branches use identical Seq2Seq architectures for latent space representation and forecasting. We combine the knowledge from the HF branch by concatenating its context vector with the main branch context vector.

Main Branch Sequence to Sequence model is made up of 2 components, an encoder and a decoder. Encoder extracts out the essential information useful for prediction from the whole input sequence by mapping it to a fixed dimensional vector in a latent space. In our case, the encoder takes in the 7-dimensional feature sequence for the past 30 minutes PM data and generates a context vector. We use a stacked 2-layer Long-short term memory (LSTM) network [29] with a hidden layer of size 512 for the encoder. LSTM has the ability to capture both long term and short term temporal dependencies from the sequence effectively instead of having a bias towards recent values and forgetting the older sequence. They also don't suffer from the vanishing gradient [30] problem. Formally, the encoder can be defined as follows:

$$\begin{aligned}
 \text{Encoder} : x_t^k, h_{t-1}^k, c_{t-1}^k &\rightarrow [h_t^k, c_t^k] \\
 f_t^k &= \sigma(W_f x_t^k + U_f h_{t-1}^k + b_f) \\
 i_t^k &= \sigma(W_i x_t^k + U_i h_{t-1}^k + b_i) \\
 o_t^k &= \sigma(W_o x_t^k + U_o h_{t-1}^k + b_o) \\
 \tilde{c}_t^k &= \tanh(W_c x_t^k + U_c h_{t-1}^k + b_c) \\
 c_t^k &= f_t^k \odot c_{t-1}^k + i_t^k \odot \tilde{c}_t^k \\
 h_t^k &= o_t^k \odot \tanh(c_t^k)
 \end{aligned} \tag{2}$$

Here, x_t^k is the input feature sequence (7 dimensional PM sequence for the 1st LSTM layer and 512 dimensional hidden layer for the 2nd layer), f_t^k , i_t^k , o_t^k are the forget, input, output gates respectively at the kth layer of the LSTM and c_t^k , h_t^k are the cell and hidden states generated at the tth time step. W_\odot, U_\odot and b_\odot are the internal weights and bias of the neural network which are learned during training via backpropagation. The hidden and cell state are initialized with 0 in the beginning, i.e., $h_0^k = c_0^k = 0$. The combination of hidden and cell state at the last time step constitutes the context vector of our network. This context vector is then passed to the decoder whose objective is to regress the conditional probability of the output BER sequence. An identical 2-layer LSTM is used for the decoder also where the hidden state and cell state are initialized with the final hidden and cell states of the encoder. The prediction is done in steps where we predict the first BER timestep for the next 15 seconds and show this value in the next iteration to the decoder LSTM as input. To prevent information compression that occurs due to the LSTM forgetting the original context vector in the future steps whilst decoding[31], we make additional connections by passing the context vector to both sides of the decoder as shown in Fig 4 .



The hidden state of the decoder along with the context vector is passed to a multi-layer perceptron (MLP) for final prediction. We also use the teacher forcing technique [32] in our architecture, where the network is shown the correct future BER values with some probability p to increase the efficiency of training. This is done as the network will throw completely random values at the beginning of training causing it to take longer total training time. Teacher forcing results in faster convergence during training and mitigate the error accumulation problem for time series regression.

HF Branch To further improve performance and make the network more sensitive to spikes, we take inspiration from some recent computer vision techniques [22, 25] and implement a parallel twin Seq2Seq branch for explicitly predicting the higher frequency (HF) components. The equations for the HF encoder are same as that of Eq. 2 apart from the fact that we additionally derive and pass the gradient of Pre FEC BER with respect to time in the input. We apply the gradient operator (∇) on the BER PM data and explicitly predict the future of this sequence also, which we call the HF sequence. The HF BER series can be calculated using the following equation:

$$\frac{\partial b_t}{\partial t} = [-1, 1] \circledast b_t \quad (3)$$

where b_t is the BER sequence and \circledast is the convolution operator. The convolution operator could be applied repeatedly to obtain even higher order derivatives, but we empirically found the usage of a single differentiation to be the most effective.

The gradient/differentiation operator separates out the high frequency components as it will only have a significant value in case of a rapid increase or decrease in the BER data. In case of normal functioning where the BER is hovering around a constant value, the differentiation operator will output values close to 0. In this branch, we pass the original 7-dimensional input features along with the historical HF BER sequence as input and train it to forecast the future HF BER sequence values. After training, the HF branch of FaS2S specifically learns to detect rapid changes in the PM data.

Fusion To pass the information generated by the HF branch constructively to our main branch, we concatenate the context vector generated by the HF branch with our primary Seq2Seq branch context vector. This allows the main branch to see the context vector generated by both the branches individually during inference. This information then gets propagated to various stages of the main decoder LSTM which significantly improves accuracy for FaS2S. We also validate this with an ablation study outlined in Sec 7. The decoder of FaS2S can be described by the following equations:

$$\begin{aligned} \text{Main Decoder: } & b_t, h^{HF}, c^{HF}, h_{t-1}^k, c_{t-1}^k \rightarrow b_{t+1} \\ f_t^k &= \sigma(W_f[b_t, h^{HF}, c^{HF}, h_{0,Dec}^k, c_{0,Dec}^k] + U_f h_{t-1}^k + b_f) \\ i_t^k &= \sigma(W_i[b_t, h^{HF}, c^{HF}, h_{0,Dec}^k, c_{0,Dec}^k] + U_i h_{t-1}^k + b_i) \\ o_t^k &= \sigma(W_o[b_t, h^{HF}, c^{HF}, h_{0,Dec}^k, c_{0,Dec}^k] + U_o h_{t-1}^k + b_o) \\ c_t^k &= \tanh(W_c[b_t, h^{HF}, c^{HF}, h_{0,Dec}^k, c_{0,Dec}^k] + U_c h_{t-1}^k + b_c) \\ c_t^k &= f_t^k \odot c_{t-1}^k + i_t^k \odot c_t^k \\ h_t^k &= o_t^k \odot \tanh(c_t^k) \\ b_{t+1} &= M.L.P(b_t, h^{HF}, c^{HF}, h_t^k, c_t^k, h_{0,Dec}^k, c_{0,Dec}^k) \end{aligned} \quad (4)$$

Here, $[\cdot, \cdot, \cdot]$ represents the concatenation of vectors. As the decoder is also a 2-layer LSTM, its equations are also similar to the encoder but there are some key differences between the two. b_t , which represents the BER at time t , is passed recursively to the decoder to predict the next value (b_{t+1}). This value is then used for the calculation of BER at the next interval (b_{t+2}). The hidden and cell state at the decoder are initialized by the hidden and cell state at the end of the encoder, i.e., $h_{0,Dec}^k = h_{Enc}^k$ and $c_{0,Dec}^k = c_{Enc}^k$. Finally, the context vectors of the main branch, HF branch and the current decoder state along with the present BER value is passed to a 3 layer MLP[33] to regress b_{t+1} .

V. TRAINING

The dataset was split into training, validation and test set in the ratio of 70:20:10 respectively. The training was carried out using pytorch framework on a Tesla T4 GPU. The best checkpoint in validation was used for calculating the final error metrics on the test set. We used *Mean Squared error (MSE)* loss from both Seq2Seq branches and the final loss function is defined as L_{reg} :

$$L_{reg} = \lambda_1(out - tar)^2 + \lambda_2(out_{HF} - tar_{HF})^2 \quad (5)$$

Here, out represents the output vector predicted by our neural network and tar represents the real BER values that are going to come in future. out_{HF} and tar_{HF} are the corresponding output and target of the HF sequence obtained using Eq. 3. Parallel to the main branch, MSE loss is also enforced for the higher frequency components and λ_1, λ_2 are hyperparameters. We empirically found that the value of $\lambda_1 = 3$ and $\lambda_2 = 1$ perform well for our network. FaS2S was trained end to end for 10 epochs with a batch size of 32 and a learning rate of 10^{-3} using Adam optimizer. 1 epoch refers to an iteration of backpropagation on the complete training set once. Each epoch in training takes about 2 hours which can be brought down by parallelizing and spreading the process among multiple GPU's. The training profile for FaS2S is shown in Fig 5. The network achieves the best results at around 8th epoch where the validation loss touches a minima of 0.172. After 8th epoch, the network starts overfitting the data and validation loss subsequently rises with further iterations.

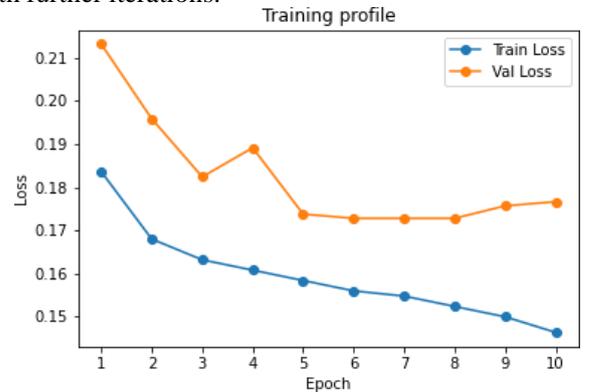


Figure 5. Evolution of training and validation loss with time

VI. RESULTS

For estimating the error, we used Mape and s Mape (symmetric Mean absolute percentage error) metrics [34] averaged over the predicted 5 minutes interval (20 timesteps). These metrics are defined as follows: -

$$Mape = \frac{1}{n} \sum_t \left| \frac{A_t - F_t}{A_t} \right| \times 100, \tag{6}$$

$$sMape = \frac{1}{n} \sum_t \left| \frac{A_t - F_t}{(A_t + F_t)/2} \right| \times 100$$

Table 2. Mape and sMape for predictions at future time steps. Error increases when we try to predict farther in the future

Metric\Time	15sec	30sec	45sec	60sec	75sec	90sec	...	240sec	255sec	270sec	285sec	300sec
Mape↓	1.113%	1.265%	1.319%	1.360%	1.391%	1.418%	...	1.600%	1.613%	1.625%	1.639%	1.651%
sMape↓	1.113%	1.265%	1.320%	1.360%	1.391%	1.418%	...	1.604%	1.617%	1.629%	1.642%	1.655%

As expected, the error accumulates when you try to predict for farther time intervals one by one using the previous predicted value as input. This causes the series to increase monotonically with time. Mape varies from 1.113% for the 1st timestep to 1.651% for the 20th timestep. sMape follows a similar distribution with Mape as our predicted values are very close with the actual values. Some of the prediction results are plotted in Fig 6 which shows the historic data the network saw during prediction along with the predicted

Here, A_t & F_t are the actual and forecasted BER values respectively at time t and n is the number of predicted timesteps (20 in our case). These values estimate the percentage error that the network has in predicting the final output. Our network was able to achieve an average Mape value of 1.485% and sMape value of 1.487% which is particularly good for a regression problem and very suitable for our use case. We also show these metrics for individual timesteps in Table 2.

output and actual target data. The model did not have any information about the future PM parameters while making the prediction and the target data has been plotted together only for easier comparison. Here you can clearly see that the model is able to figure out the important correlation from the input PM parameters and identify the peaks in BER before they happen. This can partially be attributed to our proposed methodology of incorporating HF branch in the network as we also verify using the ablation study

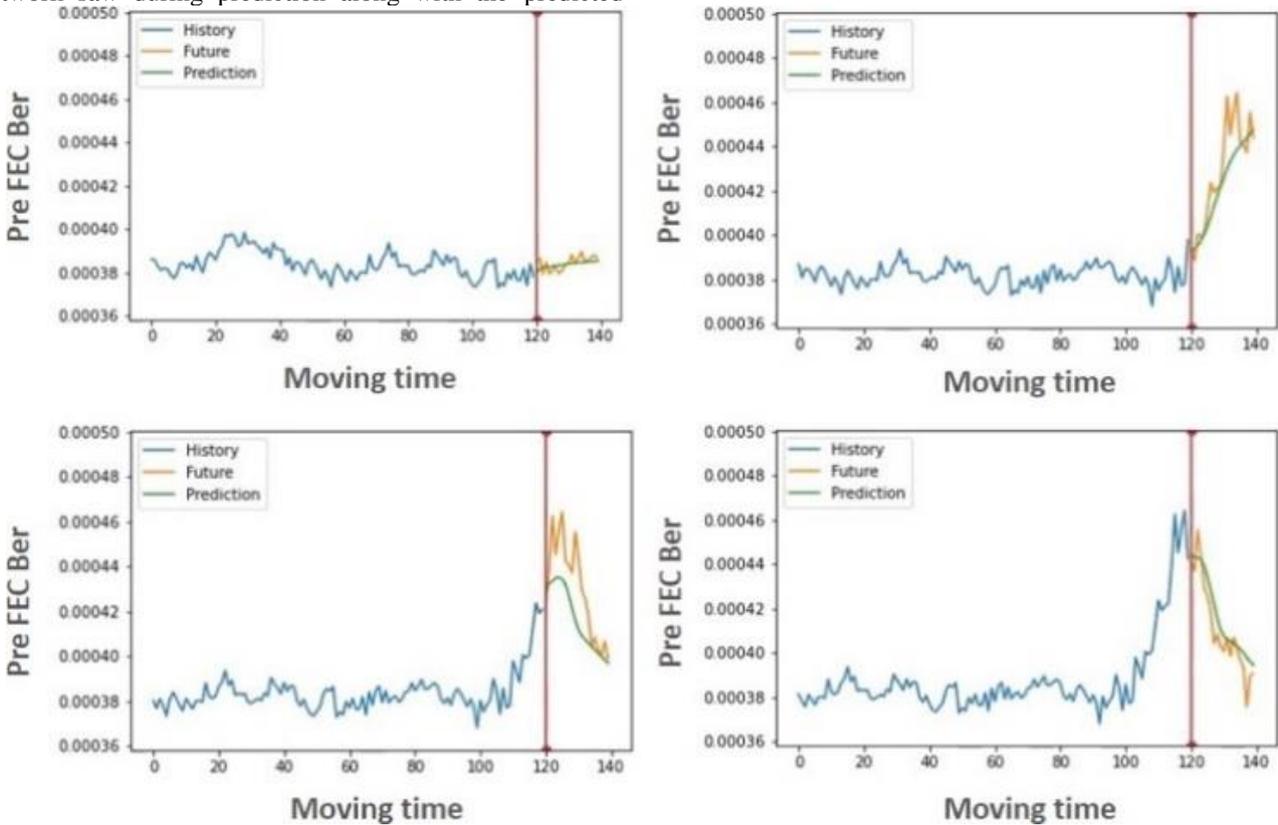


Figure 6 Qualitative results of our model showing the capability of FaS2S to detect peaks early and accurately

In Section 7. You can also find the video showing the continuous inference by FaS2S for the BER value prediction at [inference Video](#). It showcases the ability of FaS2S to perform in stable as well as unstable environments. After prediction, the information provided by FaS2S can be utilized to raise an event such as an alarm on the relevant device & propagate it to the central/distributed SDN solution. The network operator can then take an

appropriate action like manually checking for the fault or setting up automatic rerouting of the optical signal.

VII. ABLATION STUDIES

In order to further reinforce the efficacy of our method, we perform ablation studies on our model and compare it with other methods. The results are captured in Table 3.

Table 3. Using FaS2S for pro-active performance monitoring produces superior results both qualitatively and quantitatively

Method	Mape ↓	sMape ↓
ARIMA/sARIMA	-	-
Vanilla Seq2Seq	2.933%	2.996%
Seq2Seq+MLP	1.827%	1.822%
FaS2S	1.485%	1.487%

Firstly, we tried fitting ARIMA and seasonal ARIMA[35,36] statistic based models on our data using auto ARIMA[37] open source library in python. This library was not able to find suitable (p,q,d) parameters to fit the highly noisy optical networking PM data. This further validates our claim that statistical methods do not have the necessary degrees of freedom to model the complex correlations present in our data.

We also perform comparison of the performance of FaS2S in absence of MLP layer and HF branch separately. In the first method, we used a vanilla Seq2Seq[18] model and attached a linear layer with ReLu activation function to the hidden layer of the decoder. This resulted in a Mape of 2.933% and sMape of 2.996%. The performance of this model was sub-par and it ended up learning to predict a constant average value most of the time as the majority of our PM data consist of BER fluctuating in a small region. In order to increase the degrees of freedom for the model and let it generalize to more complicated scenarios, we attached a 3-layer perceptron instead of a single linear layer at the end of the decoder for the second method. This improves the accuracy by around 37% and resulted in a Mape of 1.827% and sMape of 1.822%. Even though the network was performing better at this stage, the early peak detection was still not satisfactory enough. There was also a significant number of false positives where the network predicted a wrong spike at some locations. One of the reason for these types of failure is the poor ability of the network to assess the higher frequency components in time series data. By obtaining the context vector from the prediction of the differentiation of BER sequence and concatenating it at the decoder end, FaS2S was able to mitigate this issue.

Our full architecture (FaS2S) with the HF branch was able to outperform both the ablation methods and cause an additional accuracy improvement of about 20% with a Mape of 1.485% and sMape of 1.487%. It also had good sensitivity to the sudden changes in input variables and was able to accurately map it to the future BER values.

VIII. FUTURE WORK

This approach can be generalized to do performance monitoring for the whole optical circuit instead of only at the

transponder to enable better fault localization and identify the exact site/fiber spans from where the issue is happening. We can also look at correlations between different circuits to figure out faults at a broader topology level scale. In case of fiber cuts, the model can be deployed on-premise to avoid latency delays and make quick decisions on changing to alternate optical paths. One can also extend it to a multi domain networking approach by aggregating information across IP, wireless and optical networks for a complete network management software.

IX. CONCLUSION

In this paper, we presented a data pipeline & a novel AI algorithm: Frequency aware Sequence to Sequence model, a deep learning architecture capable of utilizing higher frequency components to make precise predictions for the future Pre FEC BER in optical circuits. The network employs 7 custom selected input features for multi horizon forecasting and the output can be used for failure detection and automating network configuration management. We present a detailed qualitative and quantitative study outlining the performance of our model and also validate the same with the help of an ablation study. We believe that this work will serve as an important landmark towards the development of next-gen products for an automated optical networking management system using AI.

ACKNOWLEDGEMENT

The authors wish to thank Priyesh Bhasker, Karthik K V and Prachi Sonthalia for insights and technical support that greatly assisted our research. We also wish to thank Priya K and Ravinesh Kumar for guidance & comments on the current problem statement and its relevance for the optical networking industry.

DECLARATION

Funding/ Grants/ Financial Support	No, I did not receive.
Conflicts of Interest/ Competing Interests	No conflicts of interest to the best of our knowledge.
Ethical Approval and Consent to Participate	Work done while working at Cisco AIR Lab.
Availability of Data and Material/ Data Access Statement	Not relevant.
Authors Contributions	Both authors have contributed equally at all stages of research for this article.

REFERENCES

1. A. M. Koster, A. Zymolka, M. Jäger, and R. Hülsermann, "Demand-wise shared protection for meshed optical networks," *J. Netw. Syst. Manag.* **13**, 35–55 (2005). [CrossRef]
2. W. Du, D. Côté, C. Barber, and Y. Liu, "Forecasting loss of signal in optical networks with machine learning," *J. Opt. Commun. Netw.* **13**, E109–E121 (2021). [CrossRef]

3. A. Bakar, M. Z. Jamaludin, F. Abdullah, M. Yaacob, M. Mahdi, and M. Abdullah, "A new technique of real-time monitoring of fiber optic cable networks transmission," *Opt. Lasers Eng.* **45**, 126–130 (2007). [[CrossRef](#)]
4. G. Marra, C. Clivati, R. Luckett, A. Tampellini, J. Kronjäger, L. Wright, A. Mura, F. Levi, S. Robinson, A. Xuereb *et al.*, "Ultraprecise laser interferometry for earthquake detection with terrestrial and submarine cables," *Science*. **361**, 486–490 (2018). [[CrossRef](#)]
5. A. Sladen, D. Rivet, J.-P. Ampuero, L. De Barros, Y. Hello, G. Calbris, and P. Lamare, "Distributed sensing of earthquakes and ocean-solid earth interactions on seafloor telecom cables," *Nat. communications* **10**, 1–8 (2019). [[CrossRef](#)]
6. Y. Pointurier, "Machine learning techniques for quality of transmission estimation in optical networks," *J. Opt. Commun. Netw.* **13**, B60–B71 (2021). [[CrossRef](#)]
7. L. Velasco, P. Layec, F. Paolucci, and N. Yoshikane, "Introduction to the joint special issue on advanced monitoring and telemetry in optical networks," *J. Opt. Commun. Netw.* **13**, AMTON1–AMTON2 (2021). [[CrossRef](#)]
8. C. Natalino and P. Monti, "The Optical RL-Gym: an open-source toolkit for applying reinforcement learning in optical networks," in *International Conference on Transparent Optical Networks (ICTON)*, (2020), p. Mo.C1.1. [[CrossRef](#)]
9. C. Tremblay, S. Allogba, and S. Aladin, "Quality of transmission estimation and performance prediction of lightpaths using machine learning," in *45th European Conference on Optical Communication (ECOC 2019)*, (IET, 2019), pp. 1–3. [[CrossRef](#)]
10. C. Rottondi, L. Barletta, A. Giusti, and M. Tomatore, "Machine-learning method for quality of transmission prediction of unestablished light-paths," *J. Opt. Commun. Netw.* **10**, A286–A297 (2018). [[CrossRef](#)]
11. D. Wang, M. Zhang, Z. Li, J. Li, M. Fu, Y. Cui, and X. Chen, "Modulation format recognition and osnr estimation using cnn-based deep learning," *IEEE Photonics Technol. Lett.* **29**, 1667–1670 (2017). [[CrossRef](#)]
12. F. Locatelli, K. Christodoulouopoulos, M. S. Moreolo, J. M. Fabrega, and S. Spadaro, "Machine learning-based in-band osnr estimation from optical spectra," *IEEE Photonics Technol. Lett.* **31**, 1929–1932 (2019). [[CrossRef](#)]
13. S. Liu, D. Wang, C. Zhang, L. Wang, and M. Zhang, "Semi-supervised anomaly detection with imbalanced data for failure detection in optical networks," in *Optical Fiber Communication Conference*, (Optical Society of America, 2021), pp. Th1A–24. [[CrossRef](#)]
14. M. Zhang and D. Wang, "Machine learning based alarm analysis and failure forecast in optical networks," in *2019 24th OptoElectronics and Communications Conference (OECC) and 2019 International Conference on Photonics in Switching and Computing (PSC)*, (IEEE, 2019), pp. 1–3. [[CrossRef](#)]
15. H. Yu, L. J. Ming, R. Sumei, and Z. Shuping, "A hybrid model for financial time series forecasting—integration of ewt, arima with the improved abc optimized elm," *IEEE Access* **8**, 84501–84518 (2020). [[CrossRef](#)]
16. D. Datta, P. E. David, D. Mittal, and A. Jain, "Neural machine translation using recurrent neural network," *Int. J. Eng. Adv. Technol.* **9**, 1395–1400 (2020). [[CrossRef](#)]
17. J. Hu, X. Wang, Y. Zhang, D. Zhang, M. Zhang, and J. Xue, "Time series prediction method based on variant lstm recurrent neural network," *Neural Process. Lett.* **52**, 1485–1500 (2020). [[CrossRef](#)]
18. I. Sutskever, O. Vinyals, and Q. V. Le, "Sequence to sequence learning with neural networks," in *Advances in neural information processing systems*, (2014), pp. 3104–3112.
19. B. Zhang, D. Xiong, J. Xie, and J. Su, "Neural machine translation with gru-gated attention model," *IEEE transactions on neural networks learning systems* **31**, 4688–4698 (2020). [[CrossRef](#)]
20. C. Zhang, D. Wang, J. Jia, L. Wang, K. Chen, L. Guan, Z. Liu, Z. Zhang, X. Chen, and M. Zhang, "Potential failure cause identification for optical networks using deep learning with an attention mechanism," *J. Opt. Commun. Netw.* **14**, A122–A133 (2022). [[CrossRef](#)]
21. C. R. Morales, F. R. de Sousa, V. Brusamarello, and N. C. Fernandes, "Multivariate data prediction in a wireless sensor network based on sequence to sequence models," in *2021 IEEE International Instrumentation and Measurement Technology Conference (I2MTC)*, (IEEE, 2021), pp. 1–5. [[CrossRef](#)]
22. A. Chopra, R. Jain, M. Hemani, and B. Krishnamurthy, "Zflow: Gated appearance flow-based virtual try-on with 3d priors," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, (2021), pp. 5433–5442. [[CrossRef](#)]
23. B. Wang, H. Zheng, X. Liang, Y. Chen, L. Lin, and M. Yang, "Toward characteristic-preserving image-based virtual try-on network," in *Proceedings of the European Conference on Computer Vision (ECCV)*, (2018), pp. 589–604. [[CrossRef](#)]
24. N. Kanopoulos, N. Vasanthavada, and R. L. Baker, "Design of an image edge detection filter using the sobel operator," *IEEE J. solid-state circuits* **23**, 358–367 (1988). [[CrossRef](#)]
25. Y. Yu, F. Zhan, S. Lu, J. Pan, F. Ma, X. Xie, and C. Miao, "Wavefill: A wavelet-based generation network for image inpainting," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, (2021), pp. 14114–14123.
26. J. Yu, Z. Lin, J. Yang, X. Shen, X. Lu, and T. S. Huang, "Generative image inpainting with contextual attention," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, (2018), pp. 5505–5514.
27. X. Deng, R. Yang, M. Xu, and P. L. Dragotti, "Wavelet domain style transfer for an effective perception-distortion tradeoff in single image super-resolution," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, (2019), pp. 3076–3085. [[CrossRef](#)]
28. L. Liu, J. Liu, S. Yuan, G. Slabaugh, A. Leonardis, W. Zhou, and Q. Tian, "Wavelet-based dual-branch network for image demoiréing," in *European Conference on Computer Vision*, (Springer, 2020), pp. 86–102. [[CrossRef](#)]
29. K. Greff, R. K. Srivastava, J. Koutník, B. R. Steunebrink, and J. Schmidhuber, "Lstm: A search space odyssey," *IEEE transactions on neural networks learning systems* **28**, 2222–2232 (2016). [[CrossRef](#)]
30. Y. Hu, A. Huber, J. Anumula, and S.-C. Liu, "Overcoming the vanishing gradient problem in plain recurrent networks," *arXiv preprint arXiv:1801.06105* (2018).
31. K. Cho, B. Van Merriënboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio, "Learning phrase representations using rnn encoder-decoder for statistical machine translation," *arXiv preprint arXiv:1406.1078* (2014). [[CrossRef](#)]
32. S. Goodman, N. Ding, and R. Soricut, "Teaform: Teacher-forcing with n-grams," *arXiv preprint arXiv:2010.03494* (2020). [[CrossRef](#)]
33. H. Ramchoun, Y. Ghanou, M. Ettaouil, and M. A. Janati Idrissi, "Multi-layer perceptron: Architecture optimization and training," (2016). [[CrossRef](#)]
34. J. Tayman and D. A. Swanson, "On the validity of mape as a measure of population forecast accuracy," *Popul. Res. Policy Rev.* **18**, 299–322 (1999). [[CrossRef](#)]
35. A. A. Ariyo, A. O. Adewumi, and C. K. Ayo, "Stock price prediction using the arima model," in *2014 UKSim-AMSS 16th International Conference on Computer Modelling and Simulation*, (IEEE, 2014), pp. 106–112. [[CrossRef](#)]
36. Y.-W. Chang and M.-Y. Liao, "A seasonal arima model of tourism forecasting: The case of taiwan," *Asia Pac. journal Tour. research* **15**, 215–221 (2010). [[CrossRef](#)]
37. T. Januschowski, J. Gasthaus, and Y. Wang, "Open-source forecasting tools in python." *Foresight: The Int. J. Appl. Forecast.* (2019).

AUTHORS PROFILE



Rishabh Jain, completed his Dual Degree in B.E Hons(Computer Science) + M.Sc Hons(Physics) from Bits Pilani, Pilani Campus. He has a wide range of interests ranging to optical networking, computer vision and computational physics. He has previously worked at various research institutes and industries which include Goldman Sachs, Cisco, Nanyang Technological University, Adobe and University of Zurich. His publications include - Numerical scheme for the far-out-of-equilibrium time-dependent Boltzmann collision operator: 1D second-degree momentum discretization and adaptive time stepping (*Computer Physics Communications*, Elsevier) and Zflow: Gated appearance flow-based virtual try-on with 3d priors (*IEEE/CVF International Conference on Computer Vision*). He founded Cisco AIR Lab at Cisco Bangalore office along with Umesh Sajjanar to introduce machine learning based optimizations in optical networking elements.

Pro-active Performance Monitoring in Optical Networks using Frequency Aware Seq2Seq Model



Umesh Sajjanar, having graduated as Electronics & Communications engineer, has over 15+ years of experience in Software Development across different domains. He is currently working as a Software Engineering Manager for Cisco Optical Network Planner (CONP). He has earlier worked in the same project as a Technical Lead and served as a founding member who started Cisco AIR Labs (Cisco

Artificial Intelligence & Research Labs).