Data Augmentation to Improve Machine Learning for Optical Network Failure Management

Lareb Zar Khan(1*), João Pedro(2), Nelson Costa(2), Antonio Napoli(3), Nicola Sambo(1)

(1) Scuola Superiore Sant'Anna, via G. Moruzzi 1, 56124, Pisa, Italy (*larebzar.khan@santannapisa.it)
(2) Infinera Unipessoal Lda., Carnaxide, Portugal (3) Infinera, Munich, Germany

Abstract *A variational-autoencoder based data augmentation technique was investigated to improve the quality and increase the amount of data for optical network failure management. Augmentation provided significant performance improvement in terms of reduction in machine-learning training time for soft-failure detection (37.56%) and cause identification (66.5%). ©2022 The Author(s)*

Introduction

Machine learning (ML) is expected to offer a great potential for realizing autonomous optical networks [1]. Therefore, its applications for optical networks are being extensively studied. Numerous such applications are highlighted in [2,3], with optical network failure management (ONFM) being one of the most promising one [4]. Within ONFM, the majority of current works focuses on investigation of different ML techniques for improved failure detection, localization and cause identification, under the assumption of availability of sufficient good quality datasets. As a result of this assumption, less effort is made to improve data quality, which is a key requirement for ML. One of the ways to improve data quality is to introduce the class balance. In a balanced dataset, no class is under-represented. However, in optical networks, some failures are more common than others, resulting in uneven distribution of observations for different failures within the dataset. The training of ML models for ONFM with such imbalanced datasets may result in bias towards the majority (more common failure) class, affecting the overall performance in terms of accuracy and/or training time.

This paper focuses on data augmentation (achieved with variational-autoencoder) to address the imbalanced nature of actual experimental datasets (hereinafter referred to as *real data*). The obtained results suggest that mixing synthetic samples with real data to balance all soft-failure classes (such dataset hereinafter referred to as *modified data),* can achieve considerable performance improvement in terms of training time (more than 60% improvement achieved for failure cause identification) required to attain similar accuracy on unseen and unaugmented validation and test datasets. Moreover, results suggest that data augmentation can save resources because less faulty lightpaths need to be deployed for the collection of data.

Experimental Setup and Data Acquisition

The experimental testbed shown in Fig. 1, was used for the data collection. It consisted of four spans of single-mode-fiber, denoted by S_1, S_2, S_3, and S_4, each one 80 km long. To compensate for fiber losses, four erbium doped fiber amplifiers (EDFAs) were used, marked as A_1, A_2, A_3, and A_4. A wavelength selective switch (WSS) was placed at the end of S_2 to produce different soft-failures in the system. To realize the transmitter (Tx) and receiver (Rx), commercial coherent transponders were used.

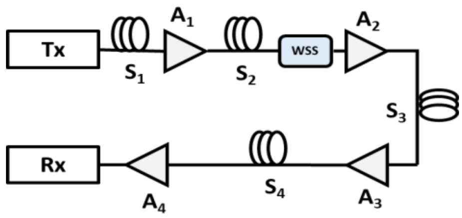

Fig. 1: Experimental Testbed setup

The input and output power levels at each amplifier, as well as bit error rate (BER) and optical signal-to-noise ratio (OSNR) at the receiver, were extracted from the testbed. For normal operation, the central frequency (f_c) of WSS was set to 192.3 THz and the attenuation and bandwidth of WSS were set to 0 dB and 37.5 GHz, respectively. Five different soft-failures were considered and their configuration details are given in Tab. 1.

Tab. 1: Considered Soft-Failures

Soft-Failure	Filter Bandwidth (GHz)	Attenuati-on (dBs)	f_c (THz)
0: Filtering	26	0	192.3
1: Attenuation	37.5	6	192.3
2: Attenuation+ Filtering	26	6	192.3
3: Change in f_c + Filtering	26	0	192.32
4: Change in f_c	37.5	0	192.32

Proposed Approach

For data augmentation, we used the variational-autoencoder (VAE) [5] based technique shown in Fig. 2.

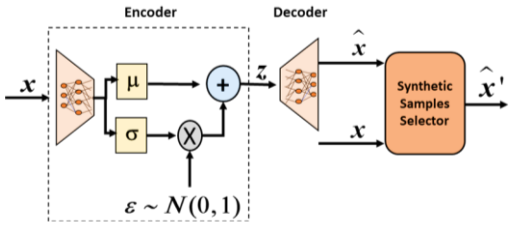

Fig. 2: Proposed Data augmentation approach; VAE followed by a synthetic sample selector

A VAE has an encoder and a decoder (both are usually neural networks) like a classical autoencoder (AE) but a VAE can be considered as the generalization of an AE to a generative model. VAE gets generative capabilities because its encoder is designed to enforce a probability distribution (gaussian in our case) on the attributes of latent space (i.e., output of encoder denoted by z in Fig. 2). During training, the VAE determines the optimal distribution parameters (μ and σ) for each latent space attribute and then the distribution is randomly sampled using reparameterization trick [6] given by Eq. (1).

$$z_j = \mu_j + \sigma_j \odot \varepsilon \qquad (1)$$

where μ_j and σ_j are the mean and standard deviation of the learned distribution for the j^{th} attribute of latent space (z_j), respectively, and ε is used for random sampling and follows a normal distribution $N(0,1)$. Such sampling enables VAE training because it makes backpropagation of error possible [6].

The VAE decoder takes a randomly sampled latent space vector as input and provides an output that should ideally be the same as the encoder's input. However, due to the imperfect training of VAE, the decoder's output is slightly different from the actual input, but it keeps the same underlying pattern as the encoder's input. In this way, if we sample the latent space multiple times, we can generate a large number of synthetic samples.

We employed a synthetic samples selector after the VAE's decoder, which selects synthetic samples that lie close to the real data in *M* dimensional space, where *M* is the number of input features. It accomplishes this by computing the Euclidean distance between each synthetic sample of any given soft-failure class and the mean of that same soft-failure class in the real data and then selects only the number of samples required to achieve a balance in the dataset.

For our analysis, the meaningful input features were BER, OSNR, and the input power at A_2. The number of attributes in the latent space was set to 2 and there was one hidden layer in encoder containing 4 neurons. The decoder had 2 neurons in the input layer, 4 in the only hidden layer and 3 in the output layer. The sigmoid activation function was used for the output layer of the VAE and where required, ReLU was used as activation function for the intermediate layers. We sampled the latent space ten times and then decoded it, which increased the available data by a factor of 10. Then, in order to obtain a modified dataset, the synthetic samples selector was employed to select the required number of synthetic samples for each soft-failure class.

Results

Fig. 3 shows the performance of the real and modified training dataset, as well as the validation data, in terms of (a) loss and (b) accuracy for soft failure detection. In general, the loss value (determined by a cost function) indicates how a model performs after each training iteration (commonly known as an epoch), and the model performance improves by minimizing this loss. On the other hand, accuracy is a more interpretable metric as it provides the ratio of correct classifications to the total number of classifications.

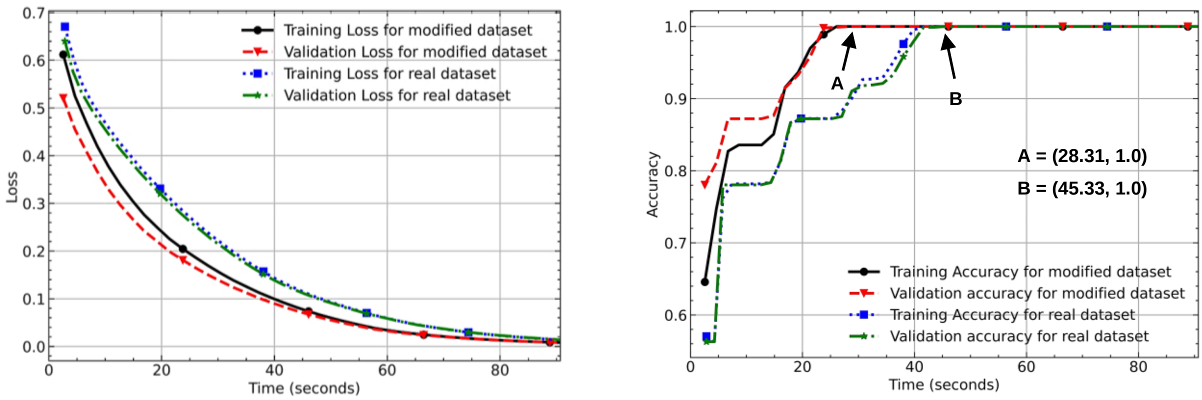

Fig. 3: Soft-Failure Detection Performance Evaluation vs time; (a) Model Loss, (b) Model Accuracy

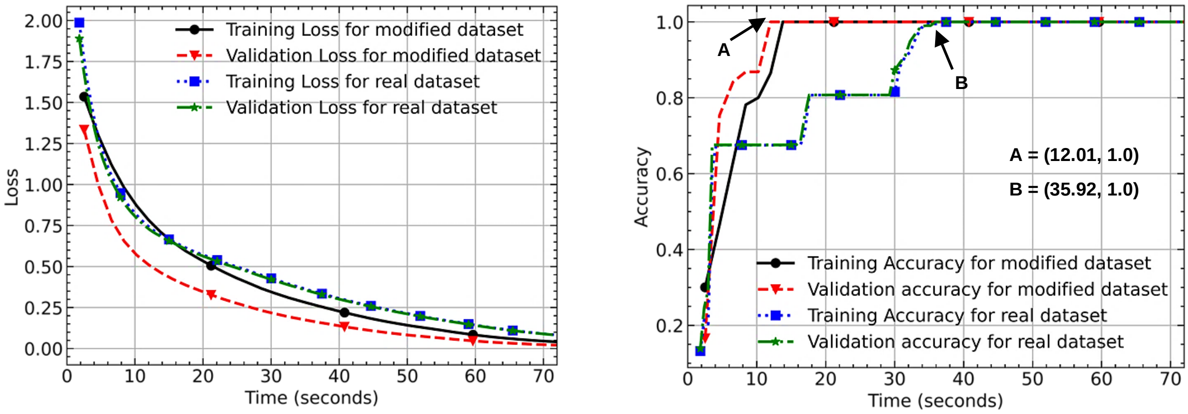

Fig. 4: Soft-Failure Cause Identification Performance Evaluation vs time; (a) Model Loss, (b) Model Accuracy

We used a simple neural network with a single hidden layer for failure detection, and as shown in Fig. 3(b), the model achieved a training and validation accuracy of 1.0 (i.e., 100%) with both the real and modified data. It is because our data was separable (i.e., non-overlapping) in 3D space; however, it is worth noting that with the modified training data, 100% validation accuracy was achieved in 28.31 seconds using our system Intel Xeon W-2255 CPU @ 3.70GHz with NVIDIA RTX A5000 GPU. But with the real data, the same accuracy was achieved after 45.33 seconds, implying a 37.56% reduction in training time with the modified data. This 37.56% lesser training time is related to the degree of imbalance between normal observations (1232) and failure observations (954). As shown, the use of a balanced dataset significantly improved the performance. Following this training, both cases attained 100% accuracy on the same test dataset. It should be noted that for a fair comparison, the validation and test datasets were not augmented, and the model used for soft-failure detection was same. Moreover, no regularization was applied to observe if modified data results in overfitting or not, and as indicated by Fig. 3, there was no overfitting.

After soft-failure failure detection, the next step is to identify its cause which is a more complex problem than the simple detection of the failure. In the real training dataset, we had 110, 74, 485, 126 and 159 observations respectively for each considered soft-failure (see Tab. 1). This imbalance was removed by adding synthetic samples which resulted in a modified dataset with 485 observations for each failure type. For this multi-class classification, another simple neural network with one hidden layer was employed, but this time there were 5 neurons in the output layer. Fig. 4 presents a comparison between the results obtained using the real and modified training datasets for soft-failure cause identification. Similar performance as for failure detection was achieved for soft-failure cause identification, but this time modified data led to the reduction of the training time by 66.5% when compared to using the real data. This is a consequence of having a class imbalance higher in this case than in the failure detection case.

This improvement in performance enabled by the balanced dataset is explained as follows: to achieve 100% accuracy, the ML model must see enough samples from each soft-failure class during training to determine the underlying pattern of that class. However, with an imbalanced training dataset, it sees an unequal number of samples in each epoch, requiring additional epochs/time to see enough samples from the underrepresented soft-failure classes. In contrast, with a balanced training dataset, the model sees enough samples of all soft-failure classes in fewer epochs, allowing it to attain 100% accuracy on unaugmented and even imbalanced validation and test datasets.

Conclusions

We investigated a variational-autoencoder based data augmentation technique to optimize ML for optical network failure management. The obtained results suggest that data augmentation can minimize the amount of resources required to obtain sufficient and suitable data for training ML models. Moreover, the training time for ML models with modified dataset (real + augmented) can be significantly reduced which is desired for the periodic re-training of ML models. Based on the degree of imbalance in the real dataset, the training time of employed ML models was reduced by 37.56% for soft-failure detection and 66.5% for the cause identification with the modified datasets.

Acknowledgements

This project has received funding from Horizon 2020's Research and Innovation Programme under grant agreement number 956713.

References

[1] R. Gu, Z. Yang and Y. Ji, “Machine Learning for Intelligent Optical Networks: A Comprehensive Survey,” *Journal of Network and Computer Applications,* vol. 157, p. 102576, 2020. DOI: 10.1016/j.jnca.2020.102576

[2] F. Musumeci, C. Rottondi, A. Nag, I. Macaluso, D. Zibar, M. Ruffini and M. Tornatore, “An Overview on Application of Machine Learning Techniques in Optical Networks,” *IEEE Communications Surveys & Tutorials,* vol. 21, no. 2, pp. 1383 - 1408, 2018. DOI: 10.1109/COMST.2018.2880039

[3] F. N. Khan, Q. Fan, C. Lu and A. P. T. Lau, “An Optical Communication's Perspective on Machine Learning and Its Applications,” *Journal of Lightwave Technology,* pp. 493 - 516, 2019. DOI: 10.1109/JLT.2019.2897313

[4] F. Musumeci, C. Rottondi, G. Corani, S. Shahkarami, F. Cugini and M. Tornatore, “A Tutorial on Machine Learning for Failure Management in Optical Networks,” *Journal of Lightwave Technology,* pp. 4125-4139, 15. DOI: 10.1109/JLT.2019.2922586

[5] D. P. Kingma and M. Welling, “Auto-Encoding Variational Bayes,” in *2nd International Conference on Learning Representations, {ICLR}*, Banff, Canada, 2014. DOI: 10.48550/arXiv.1312.6114

[6] D. P. Kingma and M. Welling, “An Introduction to Variational Autoencoders,” *Foundations and Trends® in Machine Learning,* vol. 12, no. 4, pp. 307-392, 2019. DOI: 10.1561/2200000056