Machine-Learning-Based Soft-Failure Detection and Identification in Optical Networks

Shahin Shahkarami[1]**, Francesco Musumeci**[1]**, Filippo Cugini**[2]**, Massimo Tornatore**[1]

[1]*Politecnico di Milano, Milan, Italy;* [2]*CNIT, Pisa, Italy*

[1]{*firstname.lastname*}*@polimi.it;* [2]*filippo.cugini@cnit.it*

Abstract: We develop and test several machine-learning methods to perform detection and identification of equipment failures in optical networks. Results, obtained over real BER traces, show above 98% accuracy in most cases with reasonable algorithm complexity.

OCIS codes: 060.4250 Networks; 060.4257 Networks, network survivability.

1. Introduction

Accurate monitoring of received signal quality provides a precious source of information to secure optical networks performance and to guarantee transmission quality. During network operation, several kinds of soft failures (as opposed to hard failures, where signal is totally disrupted) can affect signal quality and induce anomalies in the BER at the receiver, ultimately leading to packet losses or even service disruption. Hence, a solid mechanism for soft failure detection (i.e., recognize anomalies due to failure occurrences), localization (i.e., identify where in the network the failure occurred), and identification (i.e., understand the actual cause of the failure) is crucial, as it may be used by operators to perform traffic re-routing and rapid failure recovery [1].

New generation coherent transponders give the opportunity to monitor several parameters associated to optical signal transmission, such as Optical Signal-to-Noise-Ratio (OSNR), Q-factor, or pre-Forward Error Connection Bit Error Rate (pre-FEC BER). The amount of generated data is enormous and requires advanced data analysis techniques to extract useful information for these large data-sets. In this context, techniques from machine learning (ML) discipline are regarded as a strong candidate to address this issue, as ML enables automatized network self-configuration and fast decision-making by effectively leveraging the plethora of data that can be retrieved via network monitors. Previous works have already leveraged ML to perform monitoring [2] and even failure detection in optical networks [3]. Even though some initial research works have appeared on this topic, several questions are still pending regarding, e.g., which information is more important to be used, how often this information shall be sampled/collected by the monitors, and which ML technique (among the large set of already existing and well established tools) is better suited for soft-failure detection based on BER analysis at signal receiver.

To address some of these pending questions, in this paper we provide the following contributions: 1) we define a framework for BER anomaly detection based on monitored BER data; here we employ different ML algorithms, for which we assess the trade-off between complexity and prediction accuracy; 2) we perform a sensitivity analysis of the accuracy of the various ML techniques in order to identify the right BER sampling time, i.e., how often BER values should be collected and analyzed; 3) we propose, to the best of our knowledge for the first time in literature, a ML-based methodology to discriminate among different sources of soft failure, enabling the capability to distinguish if a BER anomaly is determined by narrow filtering (e.g., due to filters misalignment) or reduced amplification (e.g., due to amplifier malfunctioning, impacting the OSNR performance at intermediate span). In this context, ML provides a huge potential in extracting hidden patterns from BER data, while using other approaches would require the availability and analysis of additional parameters, e.g., OSNR and/or Q-factor. The whole analysis is performed using real BER traces obtained over an optical network testbed including a 380km optically amplified link and a commercial coherent polarization multiplexed quadrature phase shift keying (PM-QPSK) 100 Gb/s transmission system.

2. Soft-failure detection and identification frameworks

The proposed ML framework to perform soft-failure detection and identification is summarized in Fig. 1. The initial macro-step is *data retrieval*. First, *BER samples are periodically monitored and collected* in data-sets. Two examples of real traces of BER data (i.e., "normal" BER and anomalous BER) considered in this paper are also shown in the figure. Then, *data preprocessing* is performed to remove outliers, and the overall data-set is split into training, cross-validation and test sets taking 75%, 15% and 15% of the original data-set, respectively.

After data retrieval, the building blocks of the ML algorithms are implemented. Specifically, we first divide the samples in a set of *windows* containing several consecutive BER values. To train the ML algorithms, we set the

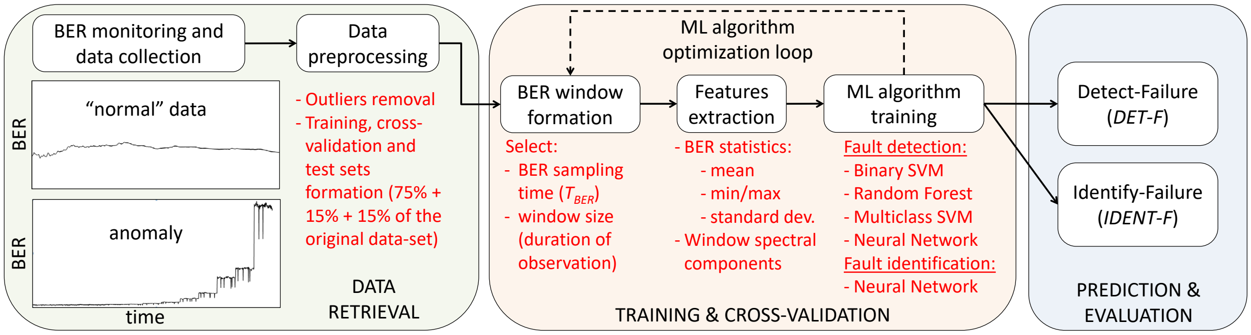


Fig. 1. Soft-failure detection and identification framework.

BER sampling time T_{BER} (i.e., the time between two consecutive BER observations in the window) and the window size (i.e., its time-duration). Note that, for a given T_{BER} and window size, different windows (i.e., different samples) may include common monitored BER values and hence they would be correlated. As an example, consider windows "a" with samples from #1 to #15 and window "b" containing BER samples #2 to #16; both windows contain 15 monitored BER observations and they share BER values #2 to #15. Then, *features are extracted* for each window, considering some statistical characteristics (i.e., minimum, maximum, mean and standard deviation of BER in the window) as well as the window's strongest spectral components, extracted by applying Fourier transform. Finally, the ML algorithms can be trained. To train the *failure detection* module (n.b., failure detection is needed to predict if a BER sequence will result into a failure or not) we use different types of ML anomaly-detection classification methods, namely Binary Support Vector Machine (SVM), Random Forest (RF), Multiclass SVM, and neural network (NN) with single hidden layer. On the other hand, to train the *failure identification* module (n.b., failure identification is needed to determine the cause of a failure), we use a NN with two hidden layers. As several parameters (e.g., number of hidden layers and nodes in the NN) can be tuned in each of the ML approaches, we run a cross-validation optimization loop to improve the classifiers (please refer to [4] for more information on how to implement these phases). While the Binary SVM is a semi-supervised classification algorithm, all the other approaches are (fully) supervised. Note that, in semi-supervised models, less training data (i.e., only the non-anomalous class) is needed to have high accuracy; conversely, in supervised approaches, sufficient data for all classes should be available to train the model properly. In our experiment, for the former case we use only "normal" BER data (i.e., BER values not resulting into a failure), whereas for the latter cases we need larger data-set, consisting of "normal" BER data and all different types of anomalies, thus impacting data storage requirements and model accuracy, which, for a given size of the data-set, is lower in comparison to semi-supervised approaches.

After training the ML modules, we test them and evaluate their performance by utilizing the *Detect-Failure (DET-F)* and *Identify-Failure (IDENT-F)* modules. *DET-F* is able to detect anomalous BER sequences, whereas the *IDENT-F* module classifies the source of anomaly, i.e., it distinguishes between filters misalignment and reduced amplification.

3. Numerical results

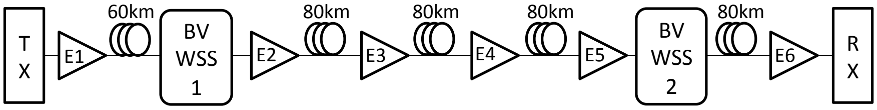
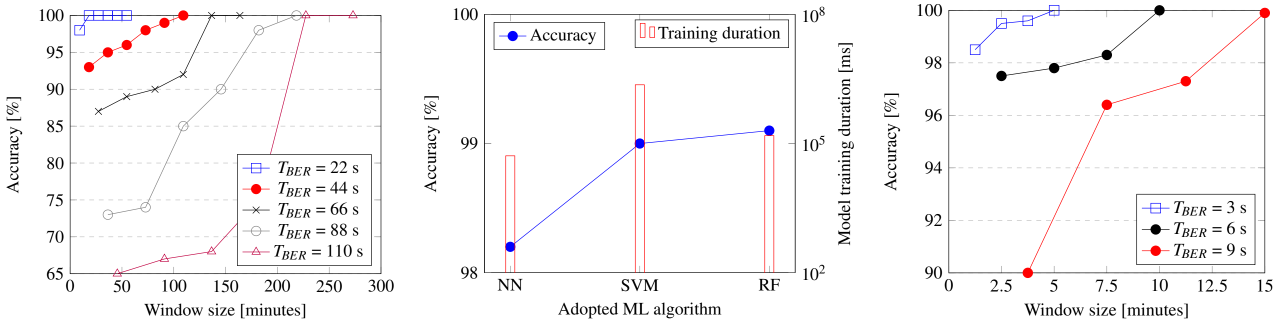


Fig. 2. Testbed setup.

To obtain numerical results we adopt the experimental setup shown in Fig. 2. Measurements were performed on an Ericsson 380km transmission system employing PM-QPSK modulation at 100 Gb/s line rate and 30.071 Gbaud. Signal is amplified through a series of 6 Erbium Doped Fibre Amplifiers (EDFA) followed by Variable Optical Attenuators (VOAs). The first Bandwidth Variable-Wavelength Selective Switch (BV-WSS 1) is configured to introduce narrow filtering or additional attenuation in intermediate span with the intent to emulate two possible impairments that cause BER degradation, i.e., filter misalignment and an undesired amplifier-gain reduction, respectively. This allowed us to gather data representing two different BER-degradation causes, over which we could train and test our *IDENT-F* module. The BV-WSS 2 does not introduce extra attenuation or filtering effects and is used only for noise reduction. We built our data-set by collecting BER samples for 24 hours, with a sampling interval of 22 seconds and 3 seconds in case of soft-failure detection and identification, respectively.

We first assess the performance of the Binary-SVM algorithm by evaluating its accuracy for different values of T_{BER}



(a) Binary-SVM accuracy vs window features.

(b) Failure detection accuracy and complexity.

(c) *IDENT-F* accuracy vs window size.

Fig. 3. Numerical results obtained for the *DET-F* (a)-(b) and *IDENT-F* (c) frameworks.

as shown in Fig. 3(a). Note that, for a given value of T_{BER}, we do not consider all possible window durations, but we stop our evaluation when we reach 100% model accuracy. From Fig. 3(a) we observe that for lower T_{BER}, a shorter window is sufficient to collect enough BER samples to optimize the performance (100% accuracy is obtained for window duration of around 18 minutes). Conversely, as T_{BER} increases, longer window duration is necessary to have acceptable performance, in order to include more BER samples and extract more significant features. For example, with $T_{BER} = 44$ s, around 73 minutes window duration is needed to have 98% accuracy.

In Fig. 3(b) we compare the accuracy (on the left y-axis of Fig. 3(b)) of the three supervised models used for soft-failure detection *DET-F*, as well as their complexity, expressed as the duration for training the model (right y-axis of Fig. 3(b)). For each of the three models we consider the values of T_{BER} and window duration providing the highest possible accuracy. The lowest complexity is obtained with the NN approach, but it also provides the worst accuracy (98.2%) among the three models. On the other hand, SVM substantially improves accuracy compared to NN, reaching 99%, but it requires longer to train the model, due to SVM algorithm complexity. The optimal compromise between accuracy and complexity is represented by RF, which provides the highest accuracy (99.1%) but it has a drastically lower computational complexity in comparison to SVM.

Finally, in Fig. 3(c) we evaluate the accuracy of our method for failure identification *IDENT-F*, using multilayer NN, for different window sizes and T_{BER}. As expected, increasing window size, *IDENT-F* accuracy increases accordingly. Moreover, for a given window duration, increasing T_{BER} reduces identification accuracy, since larger T_{BER} reduces the amount of data used to train the model. In general, window size of around 15 minutes is sufficient to provide 100% accuracy for all values of T_{BER}. In conclusion, our proposed ML approach promises to be able to identify different soft-failure sources by extracting failure-specific features over our data-set. We plan to perform further analysis to investigate which are the most relevant BER features (e.g., mean, standard deviation, some of the spectral components) to perform this identification and how our results are affected by the specific way we have emulated BER degradation.

4. Conclusion

We investigate several ML-based methods for early soft-failure detection (*DET-F*) and for the identification of the failure cause (*IDENT-F*), based on continuous monitoring of BER. We explored the trade-off between model accuracy and complexity provided by the different ML algorithms, by tuning several model parameters, such as BER sampling time and amount of BER data needed to train the models. The right tuning of parameters allows *DET-F* to reach 100% accuracy within all the considered ML approaches. For the *IDENT-F* framework, we consistently achieved 98% accuracy on our available data-set.

Acknowledgments: The work leading to these results has been supported by the European Community under grant agreement no. 761727 *Metro-Haul* project.

References

1. D. King, A. Farrel, “A PCE-based architecture for application-based network operations”, *IETF RFC 7491*, 2015.
2. S. Yan *et al.*, “Field trial of Machine-Learning-assisted and SDN-based Optical Network Planning with Network-Scale Monitoring Database”, *ECOC 2017*.
3. A. P. Vela *et al.*, “Early Pre-FEC BER Degradation Detection to Meet Committed QoS”, *OFC 2017*.
4. T. Hastie, R. Tibshirani, J. Friedman, “The Elements of Statistical Learning”, *Springer ed.*.