Monitoring and Data Analytics: Analyzing the Optical Spectrum for Soft-Failure Detection and Identification [Invited]

B. Shariati, A. P. Vela, M. Ruiz, and L. Velasco*
Optical Communications Group (GCO).
Universitat Politècnica de Catalunya (UPC), Barcelona, Spain
*lvelasco@ac.upc.edu

***Abstract*—Failure detection is essential in optical networks as a result of the huge amount of traffic that optical connections support. Additionally, the cause of failure needs to be identified so failed resources can be excluded from the computation of restoration paths. In the case of soft-failures, their prompt detection, identification, and localization make that recovery can be triggered before excessive errors in optical connections translate into errors on the supported services or even become disrupted. Therefore, Monitoring and Data Analytics (MDA) become of paramount importance in the case of soft-failures. In this paper, we review a MDA architecture that reduces remarkably detection and identification times, while facilitating failure localization. In addition, we rely on Optical Spectrum Analyzers (OSA) deployed in the optical nodes as monitoring devices acquiring the optical spectrum of outgoing links. Analyzing the optical spectrum of optical connections, specific soft-failures that affect the shape of the spectrum can be detected. A workflow consisting of machine learning algorithms, designed to be integrated in the aforementioned MDA architecture, will be studied to analyze the optical spectrum of a given optical connection acquired in a node and to determine whether a filter failure is affecting it, and in such case, what is the type of filter failure and its magnitude. Exhaustive results are presented allowing to evaluate the proposed method.**

***Keywords*—Failure Detection and Identification, Failure Magnitude Estimation, Elastic Optical Networks.**

I. Introduction

Hard failures detection at the optical layer, e.g. a fiber cut, can be easily detected, e.g. by the end transponders of optical connections. Even though the proper identification of the failed element is not an easy task, e.g. a failure in an intermediate optical amplifier, determining that the failure is in a topological element (node or link) is enough for excluding such element when restoration routes are computed [1].

However, the scenario is more difficult when we face soft-failures, such as laser drift and filter problems in the optical layer, whose presence is indirectly revealed, e.g., by observing bit error rate (BER) variations [2]. Such BER degradation, although not very high at first, could evolve toward high values and even cause disconnections. This is the very reason behind continuously monitoring the network, so such degradations can be anticipated. Nonetheless, it is not enough to monitor the BER evolution in the end transponders of optical connections, but also in intermediate nodes, so localization algorithms can determine the failed resource thus, facilitating proactive restoration strategies. In our previous work in [3], we used Optical Spectrum Analyzers (OSA) to analyze the shape of the optical spectrum of a signal to determine whether the signal was affected by a soft-failure. By installing OSAs in every optical node, such soft-failures can be easily identified, i.e., the cause of the failure is identified, and localized, being thus their output, the perfect input for restoration algorithms.

Machine learning (ML) algorithms can help in the process of detecting and identifying soft-failures. However, those algorithms should be placed closed to the network nodes aiming at reducing the amount of monitoring data to be conveyed from *Observation Points* (OP), as well as increasing the frequency of measurements, so as to reduce detection times [4]. For this very reason, the authors in [5]-[7] proposed a distributed Monitoring and Data Analytics (MDA) architecture, where data analytics capabilities are placed closed to the network nodes. OPs are configured from a centralized system to perform measurements that are immediately exported to the local MDA system, where ML algorithms are in charge of aggregating and analyzing the received data and, in case of detecting any anomaly or degradation, send a notification to the central system in charge of localizing its cause.

In this paper, we assume such distributed MDA architecture and study different ML-based methods for filter failure detection and identification. The rest of the paper is organized as follows. Section II is devoted to introduce the basic concepts about the optical spectrum and the features that allows its shape analysis. Filtering failures that might change its shape are also introduced. Next, useful ML approaches are briefly introduced and then, the distributed MDA architecture considered in this paper is summarized, so as to clearly identify where the spectrum analytics for failure detection and identification should be placed. Section III focuses on the proposed method studied in this paper; the method is based on the combined application of a classifier and failure magnitude estimators. Section IV presents representative results from realistic scenarios, where the performance of the proposed method is evaluated. Finally, Section V concludes the paper.

This work was partially supported by the EC through the METRO-HAUL (G.A. nº 761727) project, from the AEI/FEDER TWINS project (TEC2017-90097-R) and from the Catalan Institution for Research and Advanced Studies (ICREA).

978-3-903176-07-2

II. Optical Spectrum Analysis and MDA Architecture

In this section, we first introduce the features that are used to analyze the optical spectrum of a given signal and that support detecting and identifying filter failures. Next, useful ML algorithms for failure detection and identification are briefly introduced. Finally, the MDA architecture supporting data analytics distribution is presented highlighting the placement of the module for filter failure detection and identification.

A. Optical Spectrum and Filter Failures

Fig. 1 shows an example of the optical spectrum of a 100Gb/s DP-QPSK modulated signal. By inspection, we can observe that a signal is properly configured when: *i*) its central frequency is around the center of the allocated frequency slot; *ii*) its spectrum is symmetrical with respect to its central frequency; and *iii*) the effect of filter cascading is limited to a value given by the number of filters that the signal has traversed. However, when a filter failure occurs, the spectrum is distorted, and the distortion can fall into two categories: *i*) the optical spectrum is asymmetrical as a result of one or more filters are misaligned with respect to the central frequency of the slot allocated for the signal (filter shift, *FS*) and *ii*) the edges of the optical spectrum look excessively rounded compared to the expected considering the number of filters; it is a consequence of the filter's bandwidth being narrower than the slot width allocated for the signal (filter tightening, *FT*).

In order to detect the above distortions, an optical signal (which formally consists of an ordered list of frequency-power (<*f*, *p*>) pairs) can be processed to compute a number of relevant signal points that facilitate its diagnosis. Before processing an optical spectrum acquired by an OSA, the spectrum is equalized by setting its maximum power to 0 dBm. Next, a number of signal features are computed as follows:

- equalized noise level, denoted as *sig* (e.g., -60dB + equalization level),
- edges of the signal computed using the derivative of the power with respect to the frequency, denoted as ∂,
- the mean (μ) and the standard deviation (σ) of the central part of the signal computed using the edges from the derivative ($fc\_\partial \pm \Delta f$),
- a family of power levels computed with respect to μ minus $k\sigma$, denoted as $k\sigma$,
- a family of power levels computed with respect to μ minus a number of dB, denoted as *dB*.

Using these power levels, a couple of cut-off points can be generated and denoted as $f_1(\cdot)$ and $f_2(\cdot)$ (e.g., f_{1sig}, $f_{1\partial}$, f_{1dB}, $f_{1k\sigma}$). Besides, the assigned frequency slot is denoted as f_{1slot}, f_{2slot}. Combining the above, other features are computed as linear combinations of the relevant point focus on characterizing a given optical signal; they include:

- *bandwidth*, computed as $bw_{(\cdot)}=f2_{(\cdot)}-f1_{(\cdot)}$,
- *central frequency*, computed as $fc_{(\cdot)}=f1_{(\cdot)}+0.5*bw_{(\cdot)}$,
- *symmetry* with respect to a reference (frequency slot or derivatives), computed as $sym_{(\cdot)\text{-ref}}=(f1_{(\cdot)}-f1_{\text{ref}})-(f2_{\text{ref}}-f2_{(\cdot)})$.

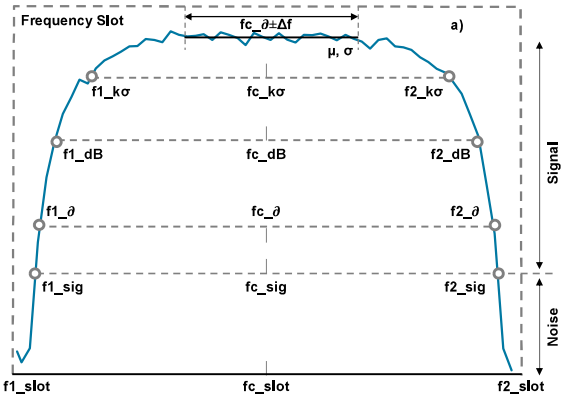

Fig. 1. Example of optical spectrum and signal features.

These features are used as input for the subsequent failure detection and identification modules.

Although relevant points have been computed from an equalized signal, note that signal distortion due to filter cascading effect has not been corrected yet. As abovementioned, this effect might induce to a wrong diagnosis of a filter problem for a normal signal. In order to overcome this drawback, we apply a *correction mask* to the measured relevant points to correct such distortions. Correction masks can be easily obtained by means of the theoretical signal filtering effects or experimental measurements taken for a distinct number of cascaded filters. Every time a diagnosis is started, the specific correction mask considering the actual number of cascading filters that the signal traverses is used to correct the relevant points.

These two different filter failures are illustrated in Fig. 2, where the solid line represents the optical spectrum of the normal signal expected at the measurement point and the solid area represents the optical spectrum of the signal with failure. Note that the expected signal is the signal used for the correction mask. In the case of filter shift, a 10 GHz shift to the right was applied (Fig. 2a), whereas the signal is affected by a 20GHz FT (Fig. 2b).

B. ML algorithms for failure detection and identification

Generally speaking, the term machine learning (ML) denotes a computer science field grouping algorithms for data analysis able to learn and make predictions from data [8]. In supervised learning, the ML algorithm is first trained with labeled data to learn a general rule that maps inputs to outputs.

Two useful ML algorithms for failure detection and identification are *classification* and *regression*. In

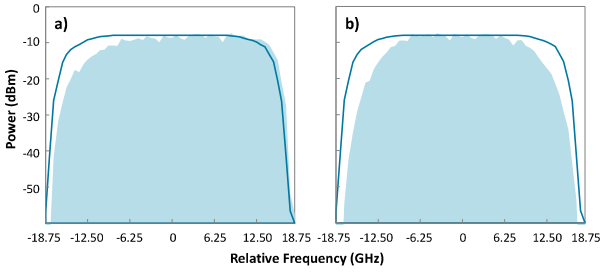

Fig. 2. Example of filter failures considered in this paper: FS (a) and FT (b).

classification, the objective is to classify unknown received data, e.g., an optical signal, and decide whether the signal belongs to the normal class, the FS class, or the FT class. In regression, the objective is to predict a behavior; e.g., regression can be used to estimate the magnitude of a failure.

Although several ML algorithms are suitable for the same output, choosing the best one is a problem-dependent decision where their performance needs to be studied for the specific case. Regarding classification, different ML methods are available in the literature, e.g., decision trees (DT) and support vector machines (SVM). A DT is a hierarchical tree structure that models the relationships between the features and the potential outcomes. DTs use a structure of branching decisions and leaves that represent the different class labels. An SVM is a binary classification technique; in the training phase, the input data is separated into groups of similar features by the computation of a boundary, called *hyperplane*, that better separates the two considered classes. As for prediction, one of the most popular algorithms is *linear regression*, which uses observations to find the best polynomial fitting for predictions.

C. *Monitoring and Data Analytics Architecture*

Let us now present the distributed MDA architecture considered in this paper, which consists of two components: the *MDA agent* and the *controller* (Fig. 3). The architecture is based on UPC's MDA platform named CASTOR [5], [6].

The MDA agent is directly connected to one or more local network nodes through an interface for configuration and another for monitoring and telemetry. The agent includes a local Knowledge Discovery from Data (KDD) module to enable local data analysis thus, reducing anomaly or failure detection times [4]; to this end, the KDD module contains KDD applications in charge of handling and processing data records. A KDD manager is the entrance point for KDD applications; it receives monitoring data records and delivers them to the corresponding KDD application. In addition, the MDA agent includes a local configuration module that enables local control loops implementation, i.e., applying local node re-configuration/re-tuning based on the results of the data analysis.

The MDA controller is a centralized system that collects monitoring data and notifications from the MDA agents and connects to the SDN controller to keep a synchronized a local copy of operational databases, e.g., topology and connections, as well as to keep it informed about any event detected in the network. The MDA controller exposes an IPFIX interface to the MDA agents so as to collect monitoring data records and notifications; received data is stored into a collected repository based on a scalable multi-master database. A process manager module is notified, and the corresponding KDD process is executed. Additionally, the MDA controller manages the configuration of the MDA agents, including KDD applications and OPs.

The role of the MDA agent is many-fold; apart from OP management, monitoring data received from active OPs can be aggregated before being sent toward the MDA controller. On the other hand, KDD applications continuously analyze

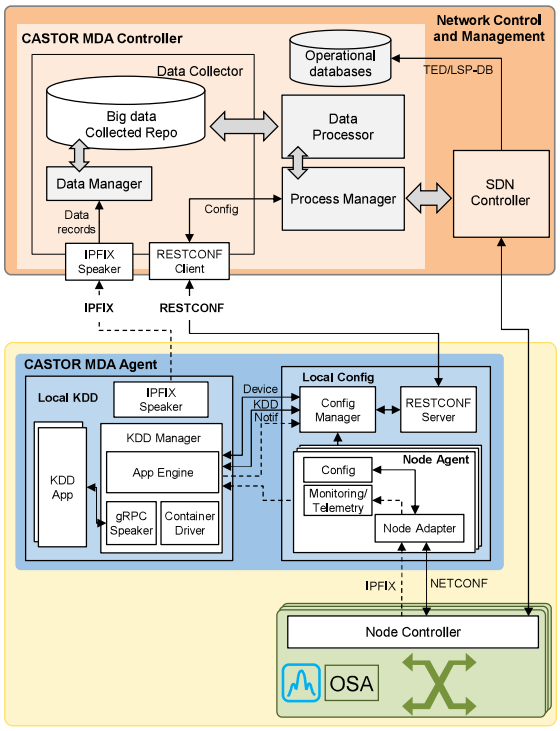

Fig. 3. CASTOR MDA architecture and interfaces.

measurements collected from OPs and send notifications toward the MDA controller in case of detecting anomalies, degradations or failures. Therefore, the MDA agent is the perfect place where the algorithm for filter failure detection and identification can be placed. In such case, the optical spectrum acquired by the local OSA(s) are periodically received through the IPFIX interface of the MDA agent. The algorithm for filter failure detection and identification receives the spectrum for every particular signal in the optical band and, in the case of detecting a failure, its class together with its estimated magnitude is reported to a hypothetical failure localization algorithm located in the MDA controller.

III. Proposed Methods for Filter Failure Detection and Identification

In this section, we define two alternative classifiers for filter failure detection and identification based on the features defined in the previous section. Additionally, we study whether transforming features would improve classification accuracy. Once the optical spectrum of a signal has been acquired in an OP, the features are extracted and corrected applying the specific correction mask that corrects filter cascading effects for the number of filters that the signal has traversed from the transmitter to the OP. Next, failure analysis can be carried out; Fig. 4 summarizes the workflow that returns the detected class of the failure (if any) and its magnitude.

The first alternative classifier is based on DTs, whereas the second one selects SVMs. Both classifiers aim at identifying whether a filter failure is affecting a connection and if so, which is the type of failure: FS or FT. In the case that a failure

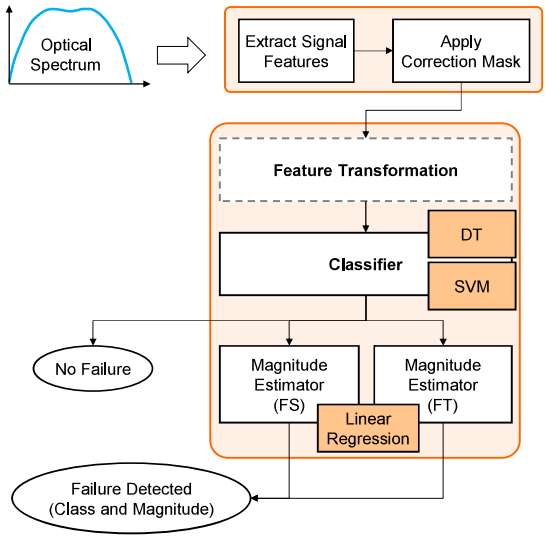

Fig. 4. Considered workflow for failure detection and identification.

has been detected, its magnitude needs to be estimated. Two filter failure magnitude estimators can be called depending on the detected failure; both are based on linear regression.

Regarding the feature transformation block, for the sake of simplicity, we consider the magnitude of the failures as additional features for training the classifiers, so we use the magnitude estimator before a failure has been detected. In this way, original features are linearly combined to create new ones that might aggregate information in the hope of improving classification accuracy.

Let us now get insight about the training process of the classifiers (see pseudocode in Table I). The algorithm receives a dataset of labeled examples that is firstly balanced by adding copies of instances from the under-represented class to have the considered classes (normal, FS, FT) equally represented (line 1 in Table I). A set of configurations that contain specific parameters for the classification algorithm selected will be used during the training process. The parameters considered to fit DTs are the number *n* of observations per leaf, for every *n* a DT model is obtained. As for SVM fitting the parameters are the degree of the polynomial kernel (*kernelDegree*) for complexity control and the cost of misclassifying (*misClassCost*) for the size of the SVM. For every configuration, a number of randomly-generated splits of the data set for training and testing will be performed. To store the goodness of each configuration, the *GoC* array will be used in the rest of the algorithm and it is firstly initialized (line 2). Next, a new dataset split is generated, where the training set is used for fitting a model for the classifier with the specific selected configuration (lines 3-7). Once a model is computed, predictions using the training and testing data set are carried out (lines 8-9); the training and testing errors between the model prediction and the actual values are stored in the *GoC* array together with the current configuration parameters (lines 10-13). Finally, the results obtained for the different configurations and training/testing data splits are evaluated to select the configuration with minimum error (line 14). Such

Table I. General classification training algorithm pseudocode

INPUT *dataset, Configs, maxSplits*
OUTPUT *model*

```
1:  dataset←balanceClassesByReplication(dataset)
2:  for each config in Configs do initialize GoC[config]
3:  for i=1..maxSplits do
4:    <trainingSet,testingSet>←randomSplit(dataset)
5:    initialize configParams
6:    for each config in Configs do
7:      model ← fit(trainingSet, config)
8:      errorTraining ←predict(model, trainingSet)
9:      errorTesting ←predict(model, testingSet)
10:     gocConfig ← GoC[config].addNew()
11:     gocConfig.configParams ← config.params
12:     gocConfig.errorTraining ← errorTraining
13:     gocConfig.errorTesting ← errorTesting
14: bestConfig←computeBestConfig(GoC)
15: return fit(dataset, bestConfig)
```

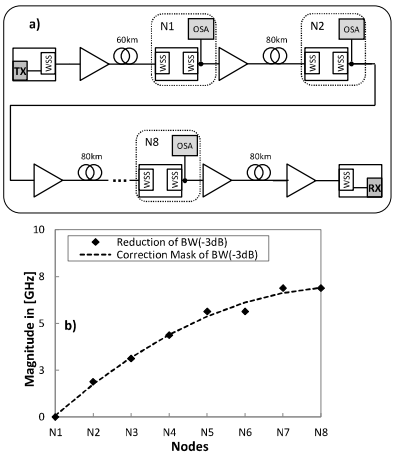

Fig. 5. VPI setup (a) and correction mask of bw_{-3dB} of the setup (b)

configuration is eventually used to fit a model using the whole dataset to improve the algorithm performance (line 15).

IV. Illustrative Results

In this section, we numerically compare the different failure identification classifiers and feature transformation described in the previous section. Let us begin by describing the transmission test-bed modeled in VPI Photonics (shown in Fig. 5a) that we use to generate the optical spectrum database required for training and testing the proposed algorithms. In the transmitter side, a 30 GBd DP-QPSK signal is generated. The signal passes through 8 single mode fiber spans. After each span, an optical amplifier compensates for the accumulated attenuation of the fiber. Each node is modeled with two 2nd order Gaussian filter emulating optical switching functionality performed by WSSs; filters bandwidth is set to 37.5 GHz, leaving 7.5 GHz as a guard band for the lightpath. Finally, the DP-QPSK signal ends in a coherent receiver that compensates for the impairments introduced throughout the transmission. One OSA per node, configured with 625 MHz resolution, is considered to monitor the optical spectrum of the lightpath. As previously discussed, a correction mask should be considered for the features affected by filter cascading (these features get reduced/increased while passing through

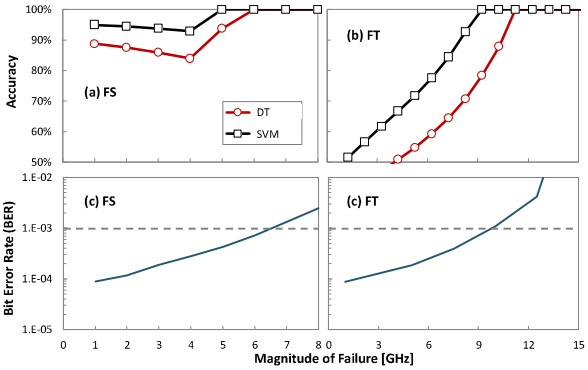

Fig. 6. Accuracy of the proposed methods for identifying FS (a) and FT (b), and BER in terms of failure magnitude of FS (c) and FT (c).

WSSs). Fig. 5b shows the amount of reduction in the magnitude of bw_{-3dB} feature and the corresponding correction mask, obtained by fitting a 2nd order polynomial.

In this study, we focus on the cases where failure happens just at the 1st node. Then, in order to emulate failure scenarios, we modify the characteristics of the 2nd WSS of the 1st node; its bandwidth and central frequency are modified to model FT and FS failures, respectively. Utilizing this setup, we collect large database of failure scenarios with different magnitude (magnitude of 1 to 15 GHz for FT and 1 to 8 GHz for FS, both with 0.25 GHz step-size.). Let us firstly focus on detecting the failure at the node where it happens, which requires one OSA per node. We use accuracy (defined as the number of correctly detected failures over the total number of the failures) as a metric to compare the performance of the different options in the workflow.

Fig. 6a and Fig. 6b show the accuracy of identifying FS and FT in terms of the magnitude of the failure, respectively; every point in Fig. 6a-b is obtained by considering all the observations belonging to that particular failure magnitude and above. This representation reveals the accuracy of the proposed classifiers (without the feature transformation block) while considering failures with magnitude above certain thresholds. For instance, the accuracy of detecting FS in a dataset comprises observations larger than 1 GHz (in our case it comprises of failures up to 8 GHz in which there are equal number of observations per each magnitude) is around ~96% for SVMs, while it hardly approaches 89% for DTs. On the other hand, the accuracy of SVMs becomes 100% for failures larger than 5 GHz, while this level of accuracy for DTs is achieved for failures larger than 6 GHz.

To better relate this accuracy to the performance of the optical transmission system, we can look into the details of the BER of the signals. Fig. 6c shows the BER evolution for increasing magnitude of FS. It is shown that the proposed methods can perfectly detect a FS problem ahead of exceeding the FEC-threshold of BER (10^{-3}). Now let us focus on the FT case. Note that the magnitude of filter tightening is defined as the difference between the ideal bandwidth of the filter (equal to 37.5 GHz) and its bandwidth during the failure. As shown in Fig. 6b, the best accuracy of the proposed classifiers for low magnitudes (below 6 GHz) is around 80% (achieved for

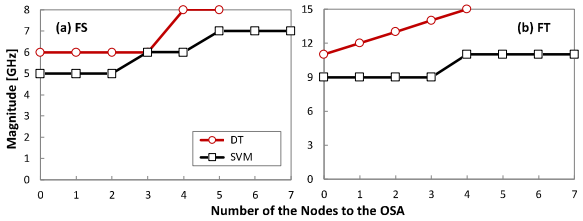

Fig. 7. Smallest failure magnitude for 100% classification accuracy vs. the OSA location for FS (a) and FT (b).

SVMs), which is due to the fact that the shape of the optical spectrum is quite similar to the normal scenario, making it very challenging for the classifier to distinguish. This is in contrast to the case of FS, which its impact is more evident from the very beginning due to its asymmetric behavior with respect to the optical spectrum. As shown in Fig. 6b, for the magnitudes above 9 GHz, exploiting SVMs, the classifier perfectly detects the failed lightpaths. However, this level of perfect accuracy is achieved for magnitudes above 10.5 GHz for the DTs. If we look at the BER performance of the system, it can be understood that the FEC-threshold limit of BER is exceeded for magnitude around 10 GHz, meaning that yet by exploiting SVMs, we are able to detect a failure ahead of disruption of the connection. Note that, the BER values reported in the paper are obtained in our VPI setup including 18 WSSs without power tuning of the components to increase the OSNR. In practice, better BER performance can be achieved with the help of more sophisticated DSP techniques and at the expense of some level of OSNR penalty [9], meaning that the detection threshold of 100% accuracy can be even further away from the FEC-limit of BER.

In the second part of the analysis, we focus on detecting the failures in some nodes after the point where the failure happens, showcasing the robustness of the proposed methods with respect to the evolution of the optical signal along the transmission line. In addition, by following this approach, the number of utilized OSAs in the network can be reduced. Fig. 7 shows the minimum magnitude after which the accuracy of classifiers remains 100% in terms of the location of OSA compared to the point that failures happen; 0 on the x-axis means that OSA is placed at the node where the failure happens (N1 in Fig. 5), while 7 means that is placed 7 nodes away from the location of the failure (N8 in Fig. 5). It can be understood that the SVM-based classifier is robust regardless of the location of the OSA and it perfectly detects the failures above a magnitude threshold where the FEC-limit of BER is not yet exceeded. Even though the DT-based classifier shows an acceptable performance for FS failures up to 3 nodes distance from the location of the failure, it fails when considering FT failures.

Once the failures are detected, filter shift estimator (FSE) and filter tightening estimator (FTE) can be launched to return the magnitude of the failures; estimators are based on linear regression. Estimated values of FS and FT with respect to their expected values are illustrated in Fig. 8a and Fig. 8b, respectively. As shown, the estimators can predict the magnitude of failures with very high accuracy, with mean

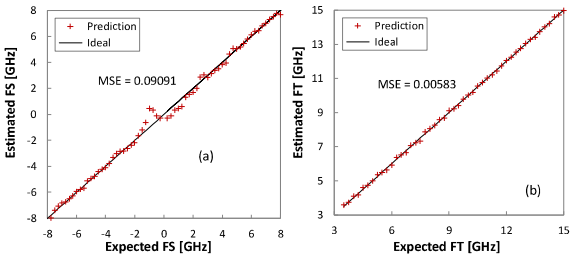

Fig. 8. Prediction accuracy of FS (a) and FT (b) estimators.

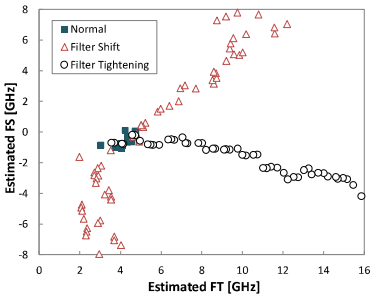

Fig. 9. Estimated FS vs estimated FT as a 2D vector space.

square error (MSE) equal to 0.09091 and 0.00583 for FSE and FTE, respectively.

In addition to the use of these estimators to explore the magnitude of the failures, they can be used in the feature transformation step as anticipated in Section III. In fact, the output of FSE and FTE can be considered as two principal components of an imaginary two-dimensional (2D) vector space as shown in Fig. 9. In such space, FS and FT failures evolve in different directions of the vector space. As illustrated in Fig. 9, the observations belonging to normal operation and the small magnitudes of the failures coincide. However, they become perfectly distinguishable as the magnitude of failures increase.

Let us evaluate the benefits of exploiting the outputs of FSE and FTE estimators as additional features for training the classifiers; Fig. 10 presents the obtained results. For the sake of conciseness, we group the magnitudes into three groups of low (L), medium (M), and high (H) magnitudes, instead of reporting all of them independently. Regarding the location of the OSAs, we report just three locations. Analyzing Fig. 10a, one can realize that the accuracy of DTs can be substantially improved, notably for low and medium magnitude, when using the estimations of FSE and FTE as new features. Additionally, it makes the classifier based on DT more robust while using OSAs far away from the location of the failures. However, it yet cannot outperform the classifier based on SVMs, even with these additional features. We also see that adding these new features does not enhance the performance of SVMs, revealing that such classifier can internally exploit the primary features to the maximum extend (Fig. 10b). Therefore, the classifier based on SVMs does not require an additional preprocessing step to generate more features; note that the magnitude of the failures are just linear combinations of primary features. Conversely, the substantial improvement seen in the DT classifier reveals that DTs cannot maximally

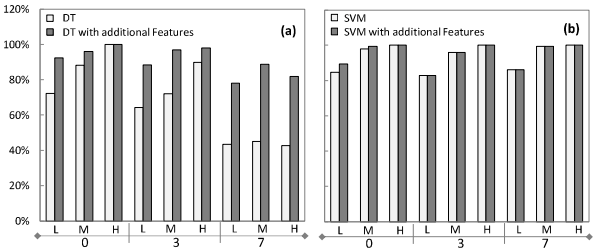

Fig. 10. Comparison of the classifiers with and without additional features.

exploit the information carried by the primary features and requires some pre-processing to grasp more information, which is a weakness of DTs compared to SVMs.

V. Concluding Remarks

In this work, we have studied the benefits of exploiting OSAs for identification and localization of filter related failures. Considering the classification methods, we have compared the performance of two different algorithms, based on DTs and SVMs, in terms of their accuracy in the detection of the failures. Additionally, we have evaluated their robustness with respect to the evolution of the optical signal along the transmission line.

The DT-based approach shows a reasonable performance if it follows a pre-processing step, aiming at generating more useful features. Otherwise, it performs weakly and lacks robustness. On the other hand, the classifiers based on SVMs have shown significant performance in detecting critical filter related failures at any point along the route of a lightpath without any pre-processing step. In addition, it benefits from very high robustness with respect to the evolution of signal along the transmission line. This robustness relaxes the requirement of deploying one OSA per node for spectrum monitoring, which in turn results in a significant reduction of the number of OSA used in the network.

References

[1] A. Castro *et al.*, "Experimental Assessment of Bulk Path Restoration in Multi-layer Networks using PCE-based Global Concurrent Optimization," IEEE Journal of Lightwave Technology (JLT), vol. 32, pp. 81-90, 2014.

[2] A. P. Vela *et al.*, "BER Degradation Detection and Failure Identification in Elastic Optical Networks," IEEE Journal of Lightwave Technology (JLT), vol. 35, pp. 4595-4604, 2017.

[3] A. P. Vela *et al.*, "Soft Failure Localization during Commissioning Testing and Lightpath Operation [Invited]," IEEE Journal of Optical Communications and Networking (JOCN), vol. 10, pp. A27-A36, 2018.

[4] A. P. Vela, M. Ruiz, and L. Velasco, "Distributing Data Analytics for Efficient Multiple Traffic Anomalies Detection," Elsevier Computer Communications, vol. 107, pp. 1-12, 2017.

[5] L. Velasco *et al.*, "An Architecture to Support Autonomic Slice Networking [Invited]," IEEE Journal of Lightwave Technology (JLT), vol. 36, pp. 135-141, 2018.

[6] Ll. Gifre *et al.*, "Autonomic Disaggregated Multilayer Networking," IEEE Journal of Optical Communications and Networking (JOCN), vol. 10, pp. 482-492, 2018.

[7] L. Velasco *et al.*, "Building Autonomic Optical Whitebox-based Networks," IEEE Journal of Lightwave Technology (JLT), 2018.

[8] C. Bishop, *Pattern Recognition and Machine Learning*, Springer-Verlag, 2006.

[9] T. Rahman *et al.*, "On the Mitigation of Optical Filtering Penalties Originating from ROADM Cascade," IEEE Photonics Technol. Letters, vol. 26, no. 2, pp. 154-157, 2014.