PCA-assisted Clustering Approaches for Soft-Failure Detection in Optical Networks [Invited]

A. N. Ribeiro[1,*], F. R. L. Lobato[1], M. F. Silva[2], A. Sgambelluri[3], L. Valcarenghi[3], L. Wosinska[4], and J. C. W. A. Costa[1]

[1] Institute of Technology, Federal University of Pará, Pará, Brazil
[2] Los Alamos National Laboratory, Los Alamos, USA
[3] Scuola Superiore Sant'Anna, Pisa, Italy
[4] Department of Electrical Engineering, Chalmers University of Technology, Gothenburg, Sweden
[*] andrei.ribeiro@itec.ufpa.br

Compiled May 27, 2026

Over the past years, the emergence of complex and bandwidth-hungry applications has charged the efforts to ensure the reliability of optical networks. Moreover, network scalability issues pose challenges as the number of optical parameters increases rapidly. In this regard, it is important to minimize the risk of optical failures by providing an autonomous and scalable failure detection approach. Hence, this paper presents a scalable and interpretable failure detection in optical networks exploiting five clustering algorithms (K-Means, Fuzzy C-Means, Gaussian Mixture Model, DBSCAN, and Mean Shift) assisted by a dimensionality reduction technique. Cluster-based approaches facilitate the physical interpretability of the failure distributions among the telemetry data by allowing their clear visualization. Meanwhile, the dimensionality reduction technique can handle large-scale telemetry data with numerous optical parameters, improving the performance of clustering algorithms, as these have limitations when dealing with high-dimensional data. The proposed approaches are evaluated based on Type I/II errors (commonly known as false positive and false negative indications, respectively). A dataset derived from an optical testbed is used to evaluate the robustness of the proposed approaches.

http://dx.doi.org/10.1364/ao.XX.XXXXXX

1. INTRODUCTION

The landscape of optical networks has witnessed the emergence of autonomous and self-aware networks that accurately handle inherent issues, such as the occurrence of failures [1, 2]. In that regard, promoting accurate and real-time fault detection is crucial for ensuring the reliability of optical transmissions. Different types of faults can affect the signal quality during the network operations. If not handled in time, faults can lead to packet loss or link disruption, directly affecting the Quality of Transmission (QoT) and further causing Service Level Agreement (SLA) violations[3, 4]. Such violations can range from delayed response times to inadequate Quality of Service (QoS). In that sense, the fulfillment of SLAs can be efficiently provided by ensuring effective monitoring of the physical layer regarding the occurrence of faults or signal degradation in the network [4–6]. Failure interpretation is also a desired feature, as it aids network operators' failure management and decision-making.

Usually, failures in optical networks can be classified into soft or hard failures. They can occur due to several factors, including components malfunctioning and aging [7], attacks in the physical layer [8], and lack of maintenance [9]. Failures that lead to the break of connections, e.g., caused by fiber cuts, directly impact the network transmissions and are categorized as hard failures. On the other hand, soft failures are typically defined as anomalies that may cause significant optical signal degradation but are not severe enough to trigger obvious alarms [2]. Soft failures can occur in the fiber and several optical devices, such as Optical Transponders (TRX), Wavelength Selective Switches (WSS), and Optical Amplifiers (OA) [10]. Those soft failures may evolve into hard failures and compromise several optical connections supporting many network services. Therefore, detecting soft-failures is essential to facilitate network maintenance and reduce operating expenses (OPEX) [11, 12].

Regarding failure detection methods, the conventional ones rely on simplified thresholds and probability models [4, 13, 14]. However, these traditional methods are ineffective, error-prone, and labor-intensive for complex and dynamic cases, which need to be conducted by experienced technicians. Thus, effective threshold-based methods can be complex as current networks are highly dynamic and can contain many optical devices [15, 16]. In this scenario, providing significant and fully automated net-

works capable of performing policy-driven self-diagnostics following the specific set of applications running over the network becomes the ultimate focus [1].

With this in mind, machine learning (ML) and artificial intelligence (AI) techniques have been increasingly investigated to address a variety of tasks related to automated optical network detection and control [13]. As using predetermined threshold methods can lead to low fault detection rates or trigger many false alarms, ML-based approaches have become a promising alternative. In this scenario, the system should be data-driven once the performance of ML is strongly dependent on the quality and amount of data for model training [17]. Also, as the network traffic dynamics may vary, data-driven strategies can guarantee an agile adaptation to the new conditions of the system. With the proper incorporation of these solutions, network operators are enabled to provide knowledge-based autonomous service provisioning [18], dealing with the dynamicity of the systems (e.g., resource allocation optimization, end-to-end QoT [19, 20], traffic profile [21–23], etc.) and guaranteeing effective detection of faults.

Several supervised learning (SL) techniques, such as Artificial Neural Networks (ANNs) [24, 26], tree-based algorithms, and Support Vector Machines (SVMs)[25, 27] can work following an SL manner. In SL-based approaches, the models need prior knowledge of data labels, containing information about what the models should provide as the correct outputs based on those labels. However, in optical networks, the datasets collected from the practical operating system always present an extremely imbalanced nature, where the volume of data under normal conditions, i.e., without faults, is much larger than the volume of data under fault conditions (if any). This imbalance leads to a slow development process, possible increase in budgetary demands, and misclassifications in models that are not trained with a considerable number of fault samples. Although ML model based on *prior-statistics* can be an option to create failure data artificially, their implementation can be critical as intensive and expensive efforts are needed for complex networks.

In this scenario, parallel to the fact that abnormal behaviors differ from normal network operation [24, 29], unsupervised learning (UL) techniques have become a promising alternative to fault management in optical networks. To mitigate the problem of imbalanced data in SL models, UL-based approaches can be applied because they only require data from the normal network conditions for model training, disregarding the acquisition of data from equipment undergoing failures or simulating failure conditions from statistical models. More specifically, these techniques can learn hidden patterns from normal data by recognizing underlying information, which means that when anomalies (failures) are fed into the model, they can be identified. Unlike the conventional supervised techniques, the ones here simplify the data collection for training purposes (as data from regular network conditions are more accessible to acquire) and accelerate model deployment. However, although properly accounting for the problem of data imbalance and interpretability, scalability concerns may limit their sole application when the network parameters become much larger in real-world scenarios [30].

Therefore, as an extension of our prior work [31], this paper introduces a semi-supervised learning approach that combines clustering algorithms with a dimensionality reduction technique to detect soft-failures in optical networks while coping with imbalanced datasets and scalability issues. Working semi-supervised means that although the model disregards the need for data under failure conditions, thereby mitigating the imbalance issue, it knows that the training data only have data from normal conditions. Hence, this approach can not be classified as a UL-based approach but can work properly. Leveraging the clustering algorithms named K-means, Fuzzy C-means (FCM), Gaussian Mixture Model (GMM), DBSCAN, and Mean Shift, the proposed approach provides failure interpretability to the operator by allowing clear visualization of data inherent characteristics and showing how the generated clusters are correlated to the data from both normal and failure conditions. Additionally, as ML models suffer from the sensitivity to high-dimensional data [32], the Principal component analysis (PCA) technique is applied, acting as a feature extractor, making the clustering models capable of handling a large volume of data commonly collected in real-world environments. The proposed approaches are evaluated in terms of false-positive (Type I errors) and false-negative (Type II errors) indications of faults.

The rest of the paper is structured as follows. Section 2 introduces closely related works proposing strategies for failure detection in optical networks. Section 3 presents the main operating principles of the five aforementioned clustering algorithms and PCA, as well as how they can be implemented together to perform optical failure detection while coping with scalability issues. Section 4 describes the optical setup used for data acquisition, data preprocessing, model fine-tuning, and the failure detection results discussion and comparison. Finally, in Section 5, we present our conclusions and the plan for future works.

2. RELATED WORKS

In recent years, the scientific community has made several efforts to develop more applicable ML algorithms for managing optical network failures. Hence, this section will discuss several works regarding ML-based approaches for failure detection.

Vela et al. proposed two different finite state machine algorithms to detect and identify the cause of several failures causing significant bit error rate (BER) degradations in optical connections [24]. Considering similar failure scenarios, the same authors also investigated ML-aided algorithms for soft failure localization [12]. Similarly, [25] compared several machine learning algorithms in complexity and accuracy to detect and identify equipment failures in optical networks. Despite effectively identifying and localizing equipment failures using BER traces, these works rely on data from failure and no-failure conditions, thereby working in a supervised learning fashion.

All the previous works use BER characteristics to realize fault management in optical networks. However, several approaches use different optical parameters for the same task. For instance, authors in [26] use a neural network-based algorithm using experimental data from optical power measurement under diverse failure modes to detect failure. Moreover, [27] proposed a soft failure localization technique based on a supervised neural network applied over telemetry data from Software-defined Networks (SDN) streams of network parameters. The proposed approach requires implementing a numerical model of the physical network to generate simulated data from failure scenarios for training. Similarly, [28] proposed a soft-failure identification and localization approach based on optical spectrum captured by optical spectrum analyzers (OSA). First, the proposed approach uses a multi-class classifier as a decision tree to detect filtering-related failures. Then, a support vector machine (SVM) binary classifier is used to predict whether the failure is due to FilterShift or TightFiltering. Similarly, [29] introduces a neural

model trained to detect the presence (or not) of a failure in an optical link based on the Optical Signal to Noise Ratio (OSNR) values from multiple lightpaths, where the model output provides a true/false response. Although efficient, this technique is limited to only indicating the presence of a failure if the model has prior knowledge about failure characteristics.

All the works mentioned above rely on SL algorithms. These algorithms require data from both failure and normal conditions to be properly trained. However, in practical optical networks, obtaining data from failure conditions is unfeasible and expensive as the design of optical links tends to be conservative and over-engineered. In this scenario, data under failure conditions become often rare in real systems, while data under normal conditions is abundant.

Therefore, recent studies have focused on UL algorithms, which only require data under normal conditions to be trained and thereby perform well with imbalanced data. For instance, [33] proposed an unsupervised approach based on generative adversarial networks (GAN) trained with electrical spectrum data to perform soft-failure detection and identification. Although the GAN model has been trained with only normal samples to perform failure detection, the proposed approach used supervised algorithms trained with failure samples to perform failure identification. Similarly, [34] performs failure management by combining a Variational Autoencoder (VAE) and a Generative Adversarial Network (GAN) to enable a threshold-less approach that relies only on Euclidean distance to distinguish between normal and failure data. However, it is not concerned about scalability. In the same sense, [35] proposed an Autoencoder-based approach to perform failure detection. Although the approach uses only normal data for the training phase, a small percentage (less than 3%) of failure data was needed to validate the model, besides not coping with scalability issues as well.

Regarding PCA-based approaches, few works leverage the rotation invariant property of PCA to handle data security concerns. For instance, [36] exploits the PCA for privacy-preserving soft-failure detection in optical networks by leveraging its ability to learn linear underlying information of the telemetry data, being able to detect abnormal behaviors (if any). For the same task, [37] proposed a disaggregated PCA approach that ensures data confidentiality by distributing the learning process within several local models. This approach prevents unauthorized access to confidential data and provides failure detection. However, although these PCA-based approaches concern data security, they disregard data scalability concerns. The same occurs in [38], where authors proposed a hybrid unsupervised/supervised fault detection framework that combines density-based clustering techniques and deep neural networks. First, the approach uses the clustering technique to learn hidden patterns from unlabeled data and classify them into clusters corresponding to fault and normal conditions. Finally, the deep neural network is fed with labeled data from the clustering technique to perform anomaly classification. Although the proposed approach works partially in an unsupervised manner, it still needs failure samples for the model training.

Furthermore, authors in [39] used a deep learning-based approach to multivariate time-series modeling for fault detection, localization, and forecasting. The proposed approach uses a large and complex architecture based on LSTNet composed of convolutional layers to guarantee scalable forecasting performance as input time series and time steps increase. Although it promotes several features regarding failure management, this work requires a dense and complex deep neural network-based approach. Also, it suffers from the neural network interpretability issue, not providing a clear interpretation of the failure data distributions to the network operator.

3. THEORETICAL BACKGROUND

This section provides the theoretical background of the algorithms exploited in this work, along with their particular application for failure detection.

A. Principal component analysis (PCA)

PCA is a widely used technique in ML and statistics for dimensionality reduction and data visualization [40]. PCA can compress the data by transforming it into a new coordinate system, where the variables are uncorrelated, orthogonal, and ordered by the amount of variance they capture. This transformation is achieved by finding the eigenvectors and eigenvalues of the covariance matrix.

In general, given a dataset X with n observations and m variables, the first step in PCA is to center the data. This is done by subtracting the mean of each variable from the respective variable values. This process ensures that the new coordinate system is aligned with the directions of maximum variance in the data rather than being centered on the means of the variables. Consequently, the arrangement of data points in this new coordinate system more accurately represents the underlying variance structure of the data. The centered data matrix is denoted as X'. After that, PCA computes the covariance matrix of the centered data X'. Each element of this matrix is calculated as the covariance between pairs of variables, and the matrix is then rearranged into a square symmetric matrix. Finally, the eigenvector decomposition is computed. The eigenvectors $\mathbf{v}$ of a covariance matrix $\mathbf{A}$ in PCA are found by solving the characteristic equation:

$$\mathbf{A}\mathbf{v} = \lambda\mathbf{v}, \tag{1}$$

where involves the following variables: $\mathbf{A}$, which represents the covariance matrix; $\mathbf{v}$, the eigenvector of the matrix $\mathbf{A}$; and λ, the eigenvalue corresponding to $\mathbf{v}$. To find the eigenvectors, one must compute the eigenvalues λ by solving the determinant equation:

$$\det(\mathbf{A} - \lambda\mathbf{I}) = 0, \tag{2}$$

where $\mathbf{I}$ is the identity matrix. After finding the eigenvalues, the corresponding eigenvectors are estimated by substitution into the characteristic equation. PCA then calculates eigenvectors $[\mathbf{v_1}, \mathbf{v_2}, ... \mathbf{v_m}]$ and eigenvalues $[\lambda_1, \lambda_2, ..., \lambda_m]$ of $\mathbf{A}$. These eigenvectors form the new basis vectors of the transformed coordinate system.

Finally, PCA selects a subset of the eigenvectors, called principal components (PC), based on the amount of variance they capture. The d PCs corresponding to the d largest eigenvalues represent the most important directions of variation in the data. In other words, the covariance matrix represents the relationships between the variables in the dataset. The eigenvectors and eigenvalues provide important information about the directions of maximum variance and the amount of variance captured by each direction, respectively. Leveraging such ability, PCA compresses the dimensions of the data into a lower one with the minimum losses possible, thereby being useful for scenarios where the number of features tends to be large.

B. K-means

The K-means algorithm is one of the most common unsupervised clustering techniques, widely used to cluster large data sets. The algorithm assigns each data sample to one of the K clusters generated by the method. K-means is a hard-clustering algorithm, which means that each sample belongs to only one cluster with the smallest Euclidean distance from the sample. The core of K-means clustering is to update the centroids by calculating the mean of the samples belonging to the cluster and repeat the relocating and updating process until convergence criteria are satisfied. The user defines the number of clusters, and the initial positions of the clusters are generally random [41].

K-means aims to minimize the Equation 3, where x is the sample belonging to the k-th cluster C and μ_k is the centroid of the cluster C_k. Some iterations are necessary until the algorithm reaches a local minimum.

$$J = \sum_{k=1}^{K} \sum_{x \in C_k} \|x - \mu_k\|^2 \tag{3}$$

Moreover, this work uses a failure indicator (FI) to analyze whether the sample was drawn from a failure condition. Different clustering approaches may follow different approaches to generate their FIs. Considering a dataset $X_{n \times m}$, with n samples of m dimensions, the K-means FIs are given using the following Equation:

$$FI(x_i, \mu_j) = \sqrt{\sum_{p=1}^{m} (x_{ip} - \mu_{jp})^2}, \tag{4}$$

where x_i is a sample from the dataset, and μ_j is the closest centroid to the sample.

C. Fuzzy C-means (FCM)

Fuzzy c-means is a centroid-based clustering algorithm similar to K-means. However, conversely, FCM is a soft-clustering algorithm [42]. This means that instead of each sample belonging to just one cluster, the sample belongs to all clusters with a certain degree of membership. This makes the equation for the centroids of each cluster a weighted average, given by:

$$c_k = \frac{\sum_{i=1}^{n} w_{ik}^p x_i}{\sum_{i=1}^{n} w_{ik}^p}, \tag{5}$$

where x_i is the i-th sample, w_{ki} is the degree of membership of the i-th sample to the k-th cluster and p is the fuzziness parameter. Membership values are initially set randomly within the range allowed by the algorithm (0 to 1) and updated after the cluster center is calculated. The updating is given by Equation 6 as follows:

$$w_{ik} = \frac{1}{\sum_{j=1}^{n} \left(\frac{\|x_i - c_k\|}{\|x_i - c_j\|} \right)^{\frac{2}{p-1}}}. \tag{6}$$

The FCM algorithm updates the membership values until Equation 7 reaches a local minimum or a pre-defined number of iterations. Note that k-means aims to minimize the same equation, just restricting the membership values.

$$J = \sum_{i=1}^{N} \sum_{k=1}^{K} w_{ik}^p \|x_i - \mu_k\|^2. \tag{7}$$

The FIs for the FCM are the same as that of K-means since both algorithms are based on the same clustering approach. For the FCM, FIs are given using the Equation 4.

D. Gaussian Mixture Model (GMM)

The Gaussian mixture model algorithm performs model-based data clustering. In that regard, multivariate finite mixture models capture the main clusters. Hence, the GMM can learn nonlinear relationships, assuming that the data can be modeled by a finite multivariate Gaussian distribution. For a GMM, each component $g(x|\theta_k)$ is represented as a Gaussian distribution,

$$g(x|\theta_k) = \frac{exp\{-\frac{1}{2}(x - \mu_k)^T \Sigma_k^{-1} (x - \mu_k)\}}{(2\pi)^{m/2} \sqrt{det(\Sigma_k)}}, \tag{8}$$

being each component denoted by the parameters, $\theta_k = \{\mu_k, \Sigma_k\}$, composed by the mean vector, μ_k and the covariance matrix, Σ_k. Thus, a GMM is completely specified by a set of parameters $\Theta = \{\alpha_1, \alpha_2, ..., \alpha_K, \theta_1, \theta_2, ..., \theta_K\}$.

Hence, a finite mixture model, $g(x|\Theta)$, is the weighted sum of $K > 1$ components $g(x|\theta_k)$ in $\mathbb{R}^m$,

$$g(x|\Theta) = \sum_{k=1}^{K} \alpha_k g(x|\theta_k), \tag{9}$$

where α_k corresponds to the weight of each component. These weights are positive $\alpha_k > 0$ with $\sum_{k=1}^{K} \alpha_k = 1$.

To estimate the GMM parameters, the expectation–maximization (EM) local search method is one of the most used [43, 44]. This method consists of two steps: i) expectation and ii) maximization. For the log-likelihood (LogL), $log(g(X|\Theta)) = log(\prod_{i=1}^{n} g(x_i|\Theta))$, the two steps are applied alternately converge to a local optimum. The performance of the EM algorithm directly depends on the choice of initial parameters Θ [45], as a poor choice of the initial parameter can result in many replications of this method during execution.

The FIs for GMM differ from those of other algorithms presented. In this case, the squared Mahalanobis distance is used to calculate the FI. Unlike the Euclidean distance, it takes into account the distribution of the data, for which the FIs are given using the following equation:

$$FI(x_i|\theta_k) = (x_i - \mu_k) \Sigma_k^{-1} (x_i - \mu_k)^T, \tag{10}$$

where x_i is a sample from the dataset and θ_k is the closest component to the sample.

E. Density-Based Spatial Clustering of Applications with Noise (DBSCAN)

DBSCAN is a density-based clustering algorithm that groups data with similar density [46]. Instead of centroids, this algorithm forms clusters by finding the core points. DBSCAN is based on two major user-chosen parameters: ϵ and MinPts, which mean the radius of the neighborhood and minimum number of points in the ϵ-neighborhood of a core point, respectively. The minimum number of neighbors within a radius ϵ (with a user-chosen distance measure) is used to estimate the minimum density level. Samples that satisfy the minimum density conditions are considered a core point. All neighbors within the ϵ radius are considered part of the same cluster as the central point. Points that are not center points and outside the radius of a center point are considered noise and do not belong to any cluster.

The FIs calculation for the DBSCAN clustering algorithm is similar to the aforementioned ones. The only difference is that this one uses the final core points found instead of centroids. DBSCAN calculates the FIs with the following equation:

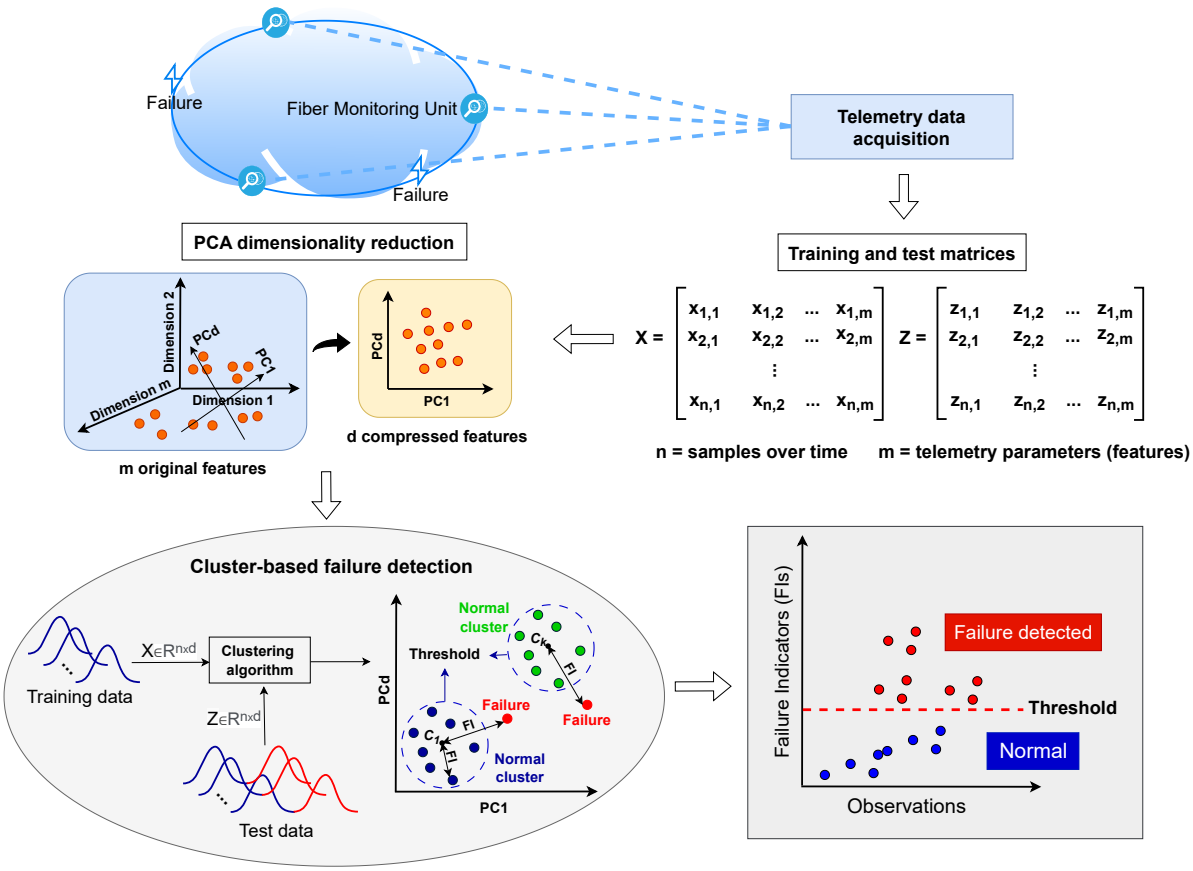


Fig. 1. Overview of the proposed approach: Steps of the PCA dimensionality reduction and cluster-based failure detection are presented, including FIs generation and the threshold usage.

$$FI(x_i, x_j) = \sqrt{(x_i - x_j)^2}, \tag{11}$$

where x_j refers to the closest core point to the dataset sample x_i.

F. Mean Shift

Mean shift is a centroid-based algorithm that clusters data points by their densities. This algorithm aims to discover "blobs" in a smooth density of samples [47]. It uses the *kernel density estimates* (KDE) to define density estimates in any smooth dimension. With a kernel such as the Gaussian kernel, a KDE requires a single user parameter, the *bandwidth*, to give the best density estimation. In that sense, Mean shift focuses on finding modes of the KDE, which refers to a local maximum of the density, by running a mean-shift iteration initialized at every data point. Hence, each mode defines one cluster, with all the points that converged to the same mode belonging to the same cluster. Candidates for centroids are updated using the mean of the points within a given region. The KDE equation and the equation of the mean-shift iteration to find modes are defined as follows:

$$p(x) = \frac{1}{N} \sum_{n=1}^{N} K\left(\left\| \frac{x - x_n}{\sigma} \right\|^2 \right), \tag{12}$$

$$f(x) = \sum_{i=1}^{n} \frac{K'\left(\left\| \frac{x - x_i}{\sigma} \right\|^2 \right)}{\sum_{i'=1}^{n} K'\left(\left\| \frac{x - x_i'}{\sigma} \right\|^2 \right)} x_i, \tag{13}$$

where $\{x_i\}_{i=1}^{n} \subset \Re^m$ are the datapoints to be clustered, σ is the bandwidth ($\sigma > 0$), K is the kernel $K(t) = e^{-t/2}$, and $K' = dK/dt$. The FIs calculation for Mean Shift is the same as for K-means or FCM.

4. PCA-ASSISTED CLUSTERING APPROACH FOR FAILURE DETECTION

Fig. 1 presents the main steps that comprise the proposed PCA-assisted clustering approach. The first step involves collecting telemetry data composed of n samples and m parameters (features) and splits into training and test datasets. Only data from normal conditions, i.e., without failures, composes the training set. Conversely, the test dataset comprises data from normal and failure conditions. The reason for using only normal data is that unsupervised models, including PCA and clustering algorithms, can learn the normal behavior of the data to later distinguish anomalous behavior.

Since the datasets are properly defined, PCA is trained and learns how to compress the m-dimensional training data into a d-dimensional dataset, with d being the number of principal components (PCs) found by PCA. This number of PCs reduces the dimensionality of the data, maintaining its maximum variance. In that case, PCA can properly transform high-dimensional telemetry data with a large number of optical parameters, typical in real-world scenarios, into a lower-dimensional dataset with much fewer parameters to be processed. Such capability

alleviates the computational cost required by ML models to process high-dimensional data while coping with the scalability issue inherent to optical communication networks with large link distances and numerous device parameters to be monitored.

Next, the clustering algorithm receives the compressed dataset spanned by PCA and trains with only data from normal conditions, as shown in the bottom left of Fig. 1. For this, the models group the normal data into clusters that could represent such type of network condition. Hence, by generating the centroid points of the normal clusters, each model can calculate the FI of every sample from the training dataset. Those clustering algorithms have different ways of generating FIs, as mentioned earlier in Section 3. FIs provide the most relevant information in the context of failure detection as these parameters can reveal the sample condition in terms of network operation, i.e., whether a sample is related to a failure condition or not. In that regard, a threshold is calculated based on the 99th percentile of the FIs derived from the training dataset. Such a parameter is further used in the testing phase to classify samples as either normal or failure.

Correspondingly, similar procedures are done in the testing phase. The exception is the dataset used in that phase, which is composed of data from both normal and failure conditions. Firstly, the trained PCA model is fed by the testing dataset, compressing it into a low-dimensional space. The compressed testing set is then fed into the clustering algorithm for FI generation. The trained clustering model computes new FIs for every sample in the testing dataset. Considering that the model had been trained with only data from normal conditions, the latest data from such conditions are expected to generate relatively small FI values.

Meanwhile, one can expect data from failure conditions to create large FI values. Hence, the FIs derived from the testing dataset are evaluated under the training-defined thresholds. If a sample eventually presents an FI value higher than the threshold, it is classified as a fault condition; otherwise, it is normal.

5. RESULTS AND DISCUSSION

This section highlights the testbed used and presents the failure detection results from the compared clustering approaches. The outlier detection performance of the compared techniques is evaluated in terms of Type I (false-positive) and Type II (false-negative) indications of failures based on a linear threshold defined for 99% confidence over the training data. This threshold value is based on the 99% percentile from the FIs derived from the training data.

A. Experimental setup and data preprocessing

The ARNO testbed is from the Tecip Institute of Scuola Superiore Sant'Anna in Pisa, Italy, and can be found in [48]. Fig. 2 shows the optical devices that compose this testbed.

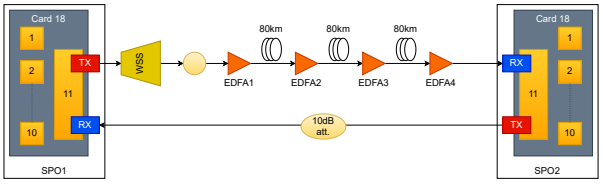


Fig. 2. Transparent optical network testbed.

It includes two Ericcson SPO 1400 devices, one Wavelength Selective Switch (WSS), and four EDFA amplifiers. At the end of the WSS, a 10 dB attenuator is installed to simulate failures. The optical link between the SPO-TX and SPO-RX consists of 3 spans of 80 km each. The data is collected with a sampling frequency of 3.5 seconds, and 5 features are used in this work, corresponding to the 4 EDFAs input powers and OSNR from SPO-RX. We evaluated several interpolation techniques, and a spline linear interpolation technique [46] presented the best fitting, totaling 13.948 samples in the final dataset. We use the OSNR values from SPO-Rx and the four EDFAs input powers as the selected features. Among them, the first 80% of the data is used for training and the last 20% for testing, as shown in Fig. 3. Failures are simulated only in the last 2 hours, corresponding to the 20% used for testing. For every 50 seconds of data collection, 10 seconds of random failures in the optical system are simulated, corresponding to 711 samples related to failures.

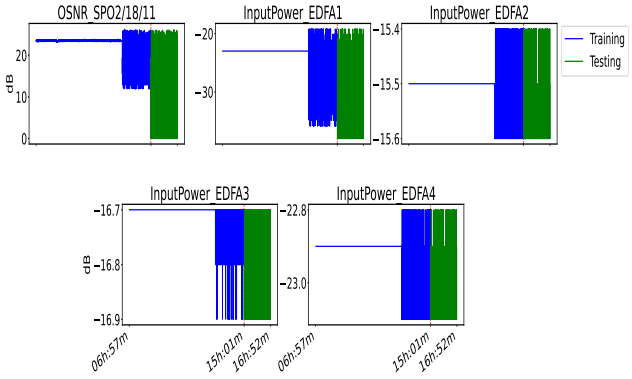


Fig. 3. Optical dataset along the training and testing sets over time.

As shown in Fig. 3, both OSNR from SPO-RX (SPO2) and EDFA1 input power present variations in the testing set. In fact, these features were expected to present such behavior as they are sensitive to failure occurrence in the network. The simulation of failures at the link between the WSS and EDFA1 directly affects the input power at this device and the OSNR calculated at the receptor. Moreover, one can note in Fig. 3 that the features' training sets show two different behaviors. The first is characterized by a line that comprises the first 6 hours of data, which represents a stationary normal condition. Conversely, the remaining 2 hours still represent data from normal conditions but with a different behavior that presents normal variations. These are generated by the attenuator at the beginning of the first link to simulate real-world network traffic. ML-based models benefit from these noise simulations as they can improve model generalization by providing information from the normal network traffic condition, which differs from the first stationary one.

B. Models fine-tuning

Throughout the PCA model training phase, it is necessary to fine-tune it to properly reduce the dimensionality of the data. The PCA performance can be optimized by finding the optimal number of PCs. Fig. 4 presents the amount of cumulative data variance retained by the PCA using different numbers of components. As shown in the figure, 2 components can properly provide dimensionality reduction, maintaining 99% of the entire data variance. Note that although the difference between the dimensionality of the original (5) and the compressed data (2)

is relatively small, in a real-world scenario where optical networks have numerous parameters, the dimensionality reduction method becomes imperative to provide model scalability. Fig. 5 shows the training dataset compressed onto the two dimensions spanned by the PCA. Approximately 84% of the entire training data is overly concentrated in the red dots presented in the figure and represents the long period of the stationary normal behavior simulated in the network.

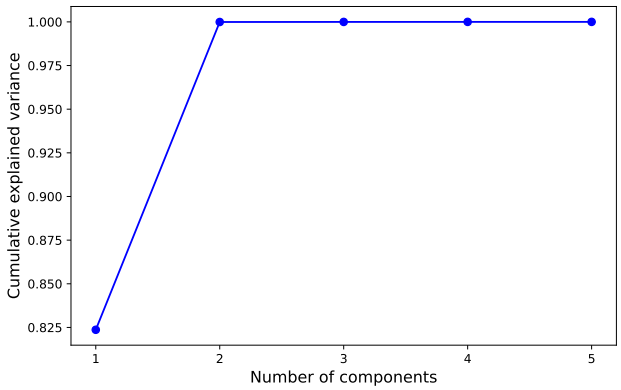


Fig. 4. Explained variance retained per number of components.

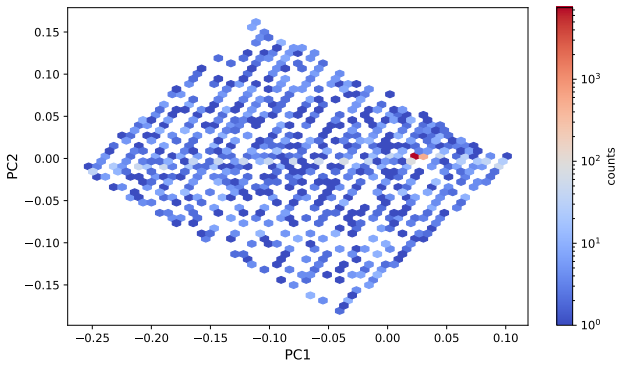


Fig. 5. Compressed training data after PCA-based dimensionality reduction.

Moreover, leveraging the 2-dimensional data compressed by PCA, the proposed clustering algorithms are trained considering their particular manner to generate the FIs, as mentioned in the previous section. We conducted several training trials to find the optimal hyperparameters for each model. For K-means, FCM, and GMM, 5 clusters presented the best failure detection results in the training phase. For DBSCAN, $\epsilon = 0.00001$ and $MinPts = 1600$ achieved the best results, while for Mean Shift, the bandwidth $\sigma = 0.05$ was the optimum value for the best results. Regarding the failure detection key parameter, the chosen confidence threshold refers to the value corresponding to the 99% percentile. This means that a level of confidence of 1% is expected at the failure detection performance. Fig. 6 shows the clusters generated from each model for all the dataset samples onto the compressed dimensions generated by PCA.

Fig. 6 demonstrates that data from failure conditions agglomerate in alternative areas to the feature space unrelated to the ones described by the normal condition. Such a fact ensures a physical interpretation of the cluster-based ML model, as each cluster representing the samples from normal condition can be easily correlated to different types of normal variations in the parameters of the optical network. Furthermore, one can note how the different densities of data presented in the dataset affected the clusterization of the models, particularly for DBSCAN. All its centroids (or core points) were positioned in the center of the massive density region in the dataset, as shown in Fig. 5. Such a fact can be expected, as DBSCAN clustering can be directly affected by the data density [46].

C. Failure detection result comparison

Table 1 introduces the failure detection results to compare the K-means, FCM, GMM, DBSCAN, and Means Shift performances. Each model is presented with their respective optimal number of clusters (or core points) (K) for further analysis regarding model complexities. Also, we compare these proposed approaches with our PCA-based approach made in previous work [37], which leveraged the same dataset for failure detection in optical networks and achieved an accuracy of 91.15%. In that sense, although all the proposed cluster-based approaches presented inferior results compared to [37], they cope with scalability issues (not approached by [37]) and provide a feasible failure detection accuracy of around 85.4% (on average).

Table 1. Failure detection results for each compared model.

Model (K)	Type I (%)	Type II (%)	Accuracy (%)
K-means (5)	14.22	0.65	85.11
FCM (5)	13.87	0.65	85.47
GMM (5)	14.13	0.58	85.28
DBSCAN (6)	14.00	4.96	81.03
Mean Shift (6)	14.45	0.55	84.98

Regarding comparing clustering models' performance, except for DBSCAN, all the models presented similar results for the evaluation metrics with an accuracy of approximately 85% (on average). Although DBSCAN achieved a similar Type I error rate, it reached approximately 5% of Type II errors. This performance could be expected by analyzing its generated clusters in Fig. 6. All the other clustering algorithms grouped the data into regions that were separated equally symmetric, providing a better model generalization. As DBSCAN focuses on high-density regions and is thereby affected by the different densities of the dataset, it probably achieved a slightly less effective performance in learning the features that characterize the normal conditions of the data. Moreover, overall, Mean Shift marginally presented the smallest rate of Type II errors. These tend to be imperative to avoid in real-world optical networks, as failure samples wrongly classified as normal samples have more impact on OPEX concerns than normal samples wrongly classified as failure samples.

Furthermore, the receiver operating characteristic (ROC) curves were plotted for each model in Fig. 7 to provide a comprehensive means of summarizing the performance of the models. They focus on the trade-off between true detection and false alarm rates. The point at the left-upper corner of the plot (0, 1) is called a perfect classification. Looking at the curves, one can note that none of the algorithms can have a perfect failure detection with a linear threshold because none of the curves go through

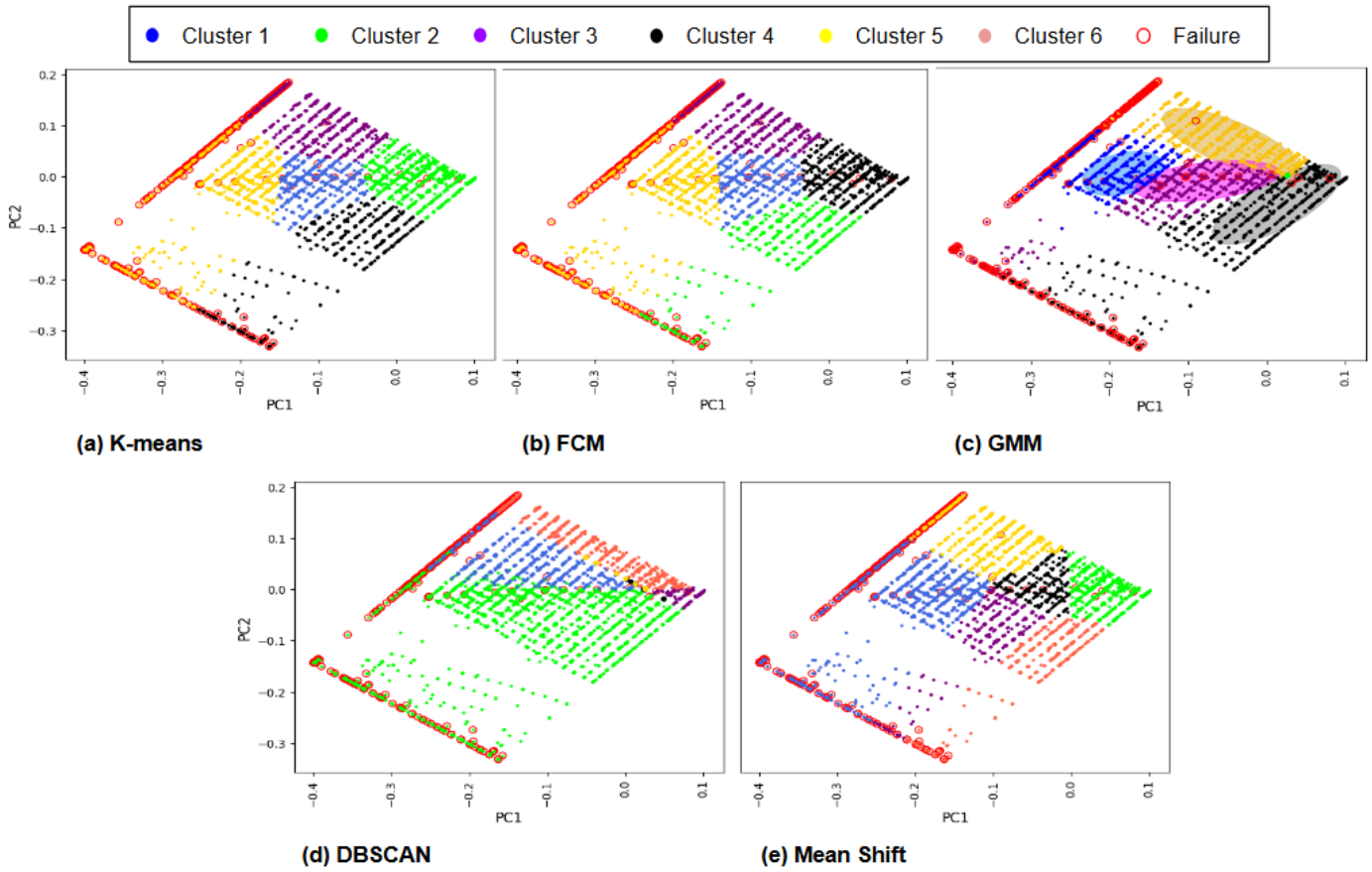


Fig. 6. Clustering solutions generated by each model onto the complete dataset. Failure samples are shown for illustrative purposes.

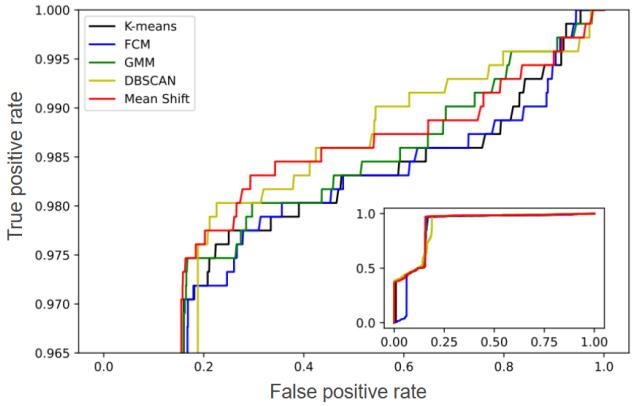


Fig. 7. ROC curves for the proposed clustering algorithms.

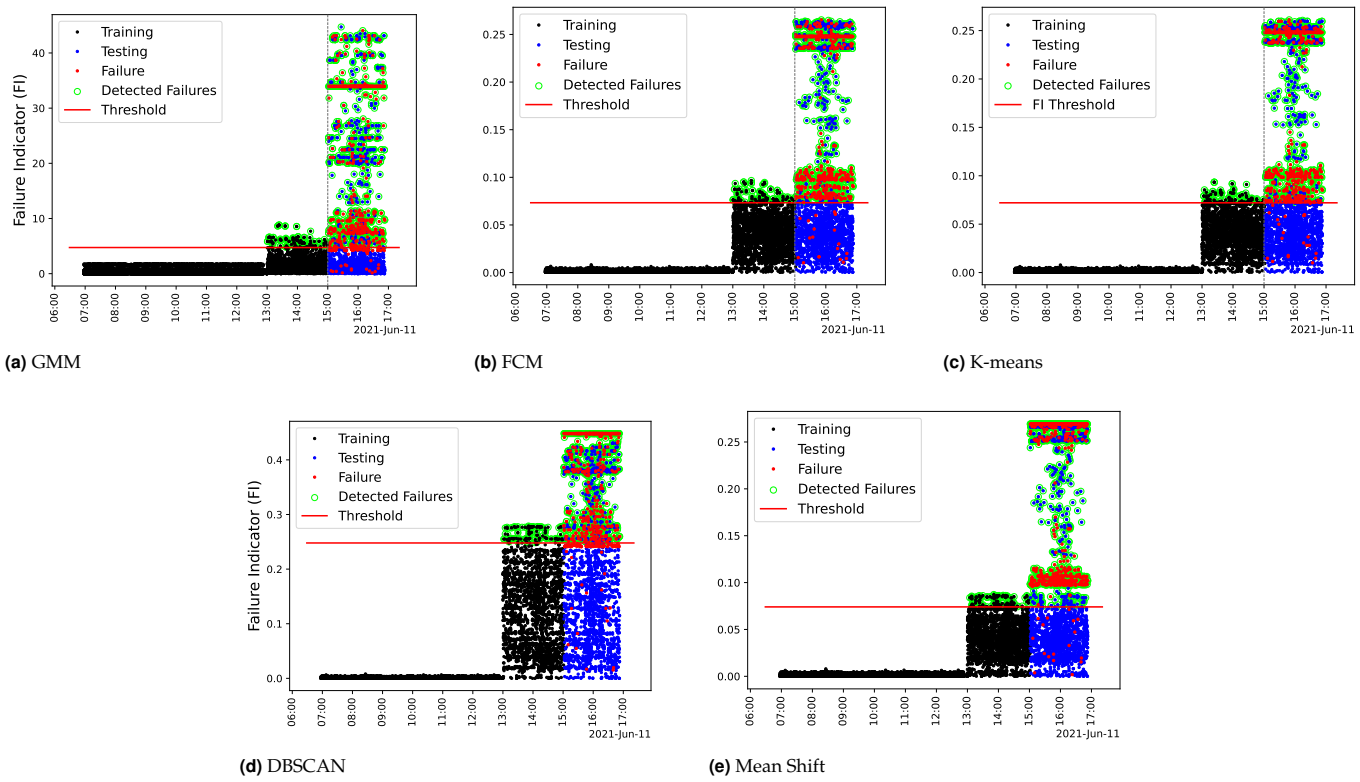
the left upper corner. Also, for levels of significance around 20% of false positive rate, Mean Shift presented a marginally better true detection rate than the other clustering algorithms, i.e., it maximizes the true detection of non-failure cases with similar performances in terms of false alarm rate.

Fig. 8 provides a clearer general view of the failure detection results of the proposed approaches. Note that some black dots appear above the threshold line in the training set for all models. These points indicate the expected margin of errors as the threshold is computed based on a 99% confidence percentile, allowing up to 1% of errors (false positives), aiming for a good trade-off of false negatives and false positives in the testing phase.

Regarding the testing set, note that the few red dots below the threshold line indicate the percentage of Type II errors. It occurs due to the simulated process of failures in the dataset. When the failure condition was simulated, although the label of the respective sample was set to failure, the system gradually changed from the normal state to the failure state, creating a few samples labeled as "failure". Although these samples have a label indicating a failure condition, they still have characteristics related to normal conditions. Thus, the model classified them as samples from normal conditions. Moreover, it's noteworthy to say that if the threshold line was set based on a smaller value than 99%, it probably could correctly detect the few failure samples below the threshold line for DBSCAN. In turn, it would generate more Type I errors as normal samples (blue dots) will appear above the threshold. Also, note the significant number of such errors in all the algorithms. These errors can be explained by analyzing the data distribution of the EDFA3 in Fig. 3. In the 10 hours of the training set, the two normal data segments have almost the same value, varying only 0.1 dBm. However, in the test set, data vary from 16.7 dBm to 16.9 dBm. As the model only learns from training set variation, this variation in the testing set leads the model to classify these normal data as failure samples wrongly.

6. CONCLUSION

In this paper, we proposed an unsupervised learning approach to detect equipment failures in optical networks using a PCA-



(a) GMM **(b)** FCM **(c)** K-means

(d) DBSCAN **(e)** Mean Shift

Fig. 8. Failure detection results over time with the threshold classification for each clustering approach.

assisted clustering algorithm, which can provide a clear interpretation of the data while coping with parameter scalability issues. Using the PCA for the particular task of dimensionality reduction provides a training dataset with a lower dimensional space, facilitating the performance of clustering models, as they do not perform well in high-dimensional datasets. Moreover, using clustering algorithms for failure detection allows a training dataset limited by data only from normal network conditions, coping with the typical imbalanced nature of the data.

For future research, there are several promising directions in extending the work. First, further studies evaluating other dimensionality reduction techniques that could provide better-compressed representations of the original dataset can be carried on. Second, additional tests leveraging a more extended dataset with different numbers of optical devices and telemetry parameters must be evaluated regarding failure detection scalability. Finally, the development of a disaggregated clustering approach leveraging Federated Learning schemes that could provide data confidentiality will be an interesting future direction.

ACKNOWLEDGMENTS

This work was partly supported by the National Council for Scientific and Technological Development (CNPq), Brazil, and partly by the project CLEVER (Project ID 101097560).

REFERENCES

1. C. Benzaid and T. Taleb, “AI-driven zero touch network and service management in 5G and beyond: Challenges and research directions,” IEEE Netw., vol. 34, no. 2, pp. 186–194, Mar./Apr. 2020, doi: 10.1109/MNET.001.1900252.
2. RAK, Jacek, et al. “Disaster resilience of optical networks: State of the art, challenges, and opportunities.” Optical Switching and Networking, v. 42, p. 100619, 2021.
3. J. Zhang and B. Mukheriee, “A review of fault management in WDM mesh networks: basic concepts and research challenges,” IEEE Netw., vol. 18, no. 2, pp. 41–48, Mar. 2004.
4. WANG, Danshi, et al. “A review of machine learning-based failure management in optical networks.” Science China Information Sciences, v. 65, n. 11, p. 1-19, 2022.
5. X. Chen et al. “Flexible availability-aware differentiated protection in software-defined elastic optical networks,” J. Lightwave Technol., vol. 33, no. 18, pp. 3872–3882, Sep.
6. X. Chen et al. “On spectrum efficient failure-independent path protection p-cycle design in elastic optical networks,” J. Lightw. Technol., vol. 33, no. 17, pp. 3719–3729, Sep. 2015.
7. Z. Cheng et al. “T-trail: Link failure monitoring in software-defined optical networks,” J. Opt. Commun. Netw., vol. 10, no. 4, pp. 344–352, Apr. 2018.
8. C. Mas, I. Tomkos, and O. Tonguz, “Failure location algorithm for transparent optical networks,” IEEE J. Sel. Areas Commun., vol. 23, no. 8, pp. 1508–1519, Aug. 2005.
9. R. Rejeb, M. Leeson, and R. Green, “Fault and attack management in all-optical networks,” IEEE Commun. Mag., vol. 44, no. 11, pp. 79–86, Nov. 2006.
10. Mayer et al. “Soft failure localization using machine learning with SDN-based network-wide telemetry.” 2020 European Conference on Optical Communications (ECOC). IEEE, 2020.
11. Barzegar, Sima, et al. “Soft-failure localization and device working parameters estimation in disaggregated scenarios.” Optical Fiber Communication Conference. Optica Publishing Group, 2020.
12. A. Vela et al. “Soft failure localization during commissioning testing and lightpath operation,” J. Opt. Commun. Netw., vol. 10, no. 1, pp. A27–A36, Jan. 2018.
13. Musumeci, Francesco, et al. “An overview on application of machine

learning techniques in optical networks." IEEE Communications Surveys & Tutorials 21.2 (2018): 1383-1408.
14. Musumeci, Francesco, et al. "A tutorial on machine learning for failure management in optical networks." Journal of Lightwave Technology 37.16 (2019): 4125-4139.
15. M. Zhang et al. "Dynamic and adaptive bandwidth defragmentation in spectrum-sliced elastic optical networks with time-varying traffic," J. Lightw. Technol., vol. 32, pp. 1014–1023, Mar. 2014.
16. Y. Pointurier, "Design of low-margin optical networks," J. Opt. Commun. Netw., vol. 9, no. 1, pp. A9–A17, 2017.
17. M. Xie et al. "AI-driven closed-loop service assurance with service exposures," in Proc. Eur. Conf. Netw. Commun (EuCNC), 2020, pp. 265–270, doi: 10.1109/EuCNC48522.2020.9200943.
18. X. Chen et al. "Knowledge-based autonomous service provisioning in multi-domain elastic optical networks," IEEE Commun. Mag., vol. 56, no. 8, pp. 152–158, Aug. 2018.
19. R. Proietti et al. "Experimental demonstration of machine-learning-aided QoT estimation in multi-domain elastic optical networks with alien wavelengths," J. Opt. Commun. Netw., vol. 11, no. 1, pp. A1–A10, Jan. 2019.
20. S. Oda et al. "A learning living network with open ROADMs," J. Lightw. Techn., vol. 35, no. 8, pp. 1350–1356, Apr. 2017.
21. X. Chen et al. "Leveraging deep learning to achieve knowledge-based autonomous service provisioning in broker-based multi-domain SD-EONs with proactive and intelligent predictions of multi-domain traffic," in Proc. of ECOC, Sept. 2017, pp. 1–3.
22. B. Li et al. "Deep-learning-assisted network orchestration for on-demand and cost-effective vNF service chaining in inter-DC elastic optical networks," J. Opt. Commun. Netw., vol. 10, pp. D29–D41, Oct. 2018.
23. J. Guo and Z. Zhu, "When deep learning meets inter-datacenter optical network management: Advantages and vulnerabilities," J. Lightw. Technol., vol. 36, no. 20, pp. 4761–4773, Oct 2018.
24. A. Vela et al. "BER degradation detection and failure identification in elastic optical networks," J. Lightw. Technol., vol. 35, no. 21, pp. 4595–4604, Nov. 2017.
25. Shahkarami et al. "Machine-learning-based soft-failure detection and identification in optical networks," in Proc. of OFC, Mar. 2018, pp. 1–3.
26. Rafique et al. "Cognitive assurance architecture for optical network fault management," J. Lightw. Technol., vol. 36, no. 7, pp. 1443–1450, Apr. 2018.
27. Mayer et al. "Machine-learning-based soft-failure localization with partial software-defined networking telemetry," J. Opt. Commun. Netw., vol. 13, no. 10, pp. E122–E131, 2021, doi: 10.1364/JOCN.424654.
28. Velasco L, Shariati B, Vela A P, et al. "Learning from the optical spectrum: Soft-failure identification and localization." In: Proceedings of 2018 Optical Fiber Communications Conference and Exhibition (OFC), San Diego, 2018. 1–3.
29. D. Y. Shimizu, K. S. Mayer, J. A. Soares, and D. S. Arantes, "A deep neural network model for link failure identification in multi-path ROADM based networks," in Proc. Photon. North (PN), 2020, p. 1, doi: 10.1109/PN50013.2020.9166978.
30. Furdek, Marija, et al. "Optical network security management: requirements, architecture, and efficient machine learning models for detection of evolving threats", Journal of Optical Communications and Networking 13.2 (2021): A144-A155.
31. Ribeiro, A. N., et al. "PCA-Assisted Fuzzy Clustering Approach for Soft-Failure Detection in Optical Networks." 2024 International Conference on Optical Network Design and Modeling (ONDM). IEEE, 2024.
32. S. A. Shah and V. Koltun, "Deep continuous clustering", arXiv:1803.01449 (2018).
33. Lun, Huazhi, et al. "GAN based soft failure detection and identification for long-haul coherent transmission systems." 2021 Optical Fiber Communications Conference and Exhibition (OFC). IEEE, 2021.
34. Kruse, Lars E., et al. "Experimental investigation of machine-learning-based soft-failure management using the optical spectrum." Journal of Optical Communications and Networking 16.2 (2024): 94-103.
35. Liu, SongLin, et al. "Semi-supervised anomaly detection with imbalanced data for failure detection in optical networks." Optical Fiber Communication Conference. Optica Publishing Group, 2021.
36. Silva, Moises Felipe, et al. "Confidentiality-preserving machine learning algorithms for soft-failure detection in optical communication networks." Journal of Optical Communications and Networking 15.8 (2023): C212-C222.
37. Sales, R. F., et al. "Disaggregated Confidentiality-Preserving Scheme for Fault Detection in Optical Networks." 2024 Optical Fiber Communications Conference and Exhibition (OFC). IEEE, 2024.
38. Chen, Xiaoliang, et al. "On cooperative fault management in multi-domain optical networks using hybrid learning." IEEE Journal of Selected Topics in Quantum Electronics 28.4: Mach. Learn. in Photon. Commun. and Meas. Syst. (2022): 1-9.
39. Silva, Moisés Felipe, et al. "Learning long-and short-term temporal patterns for ML-driven fault management in optical communication networks." IEEE Transactions on Network and Service Management 19.3 (2022): 2195-2206.
40. Kramer, Mark A. "Nonlinear principal component analysis using autoassociative neural networks." AIChE journal 37.2 (1991): 233-243.
41. Ahmed, Mohiuddin, Raihan Seraj, and Syed Mohammed Shamsul Islam. "The k-means algorithm: A comprehensive survey and performance evaluation." Electronics 9.8 (2020): 1295.
42. Bezdek, James C. "Pattern recognition with fuzzy objective function algorithms." Springer Science & Business Media, 2013.
43. G. McLachlan and D. Peel, "Finite Mixture Models," Wiley Series in Probability and Statistics (Wiley, 2004).
44. Dempster, Arthur P., Nan M. Laird, and Donald B. Rubin. "Maximum likelihood from incomplete data via the EM algorithm." Journal of the royal statistical society: series B (methodological) 39.1 (1977): 1-22.
45. M. A. T. Figueiredo and A. K. Jain, "Unsupervised learning of finite mixture models," in IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 24, no. 3, pp. 381-396, March 2002, doi: 10.1109/34.990138.
46. Khan, Kamran, et al. "DBSCAN: Past, present and future." The fifth international conference on the applications of digital information and web technologies (ICADIWT 2014). IEEE, 2014.
47. Carreira-Perpinán, Miguel A. "A review of mean-shift algorithms for clustering." arXiv preprint arXiv:1503.00687 (2015).
48. InRete Lab, 2021, "Optical Failure Dataset", Scuola Superiore Sant'Anna. [Online]. Available: https://github.com/NetworkAndServices/optical-failure-dataset
49. Junninen, Heikki, et al. "Methods for imputation of missing values in air quality data sets." Atmospheric environment 38.18 (2004): 2895-2907.