Confidentiality-preserving Machine Learning Algorithms for Soft-failure Detection in Optical Communication Networks

MOISES FELIPE SILVA[1,*], ANDREA SGAMBELLURI[2], ALESSANDRO PACINI[2], FRANCESCO PAOLUCCI[3], ANDRE GREEN[1], DAVID MASCARENAS[1], AND LUCA VALCARENGHI[2]

[1] Los Alamos National Laboratory, Los Alamos, USA
[2] Scuola Superore Sant'Anna, Pisa, Italy
[3] CNIT, Pisa, Italy
[*] Corresponding author: mfelipe@lanl.gov

Compiled January 19, 2025

Automated fault management is at the forefront of next-generation optical communication networks. The increase in complexity of modern networks has triggered the need for programmable and software-driven architectures to support the operation of agile and self-managed systems. In these scenarios the ETSI zero-touch network and service management (ZSM) approach is imperative. The need for machine learning algorithms to process the large volume of telemetry data brings safety concerns as distributed cloud-computing solutions become the preferred approach for deploying reliable communication network automation. This paper's contribution is twofold. First, we propose a simple yet effective method to guarantee the confidentiality of the telemetry data based on feature scrambling. The method allows the operation of third-party computational services without direct access to the full content of the collected data. Additionally, the effectiveness of four unsupervised machine learning algorithms for soft-failure detection is evaluated when applied to the scrambled telemetry data. The methods are based on factor analysis, principal component analysis, nonlinear principal component analysis, and singular value decomposition. Most dimensionality reduction algorithms have the common property that they can maintain similar levels of fault classification performance while hiding the data structure from unauthorized access. Evaluations of the proposed algorithms demonstrate this capability.

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

1. INTRODUCTION

Communication networks, including optical networks, are currently designed with the aim to provide open and online monitoring data processing and analysis. Software defined networking (SDN) is constantly evolving to support autonomous optical communication networks in which network awareness is implemented in a closed loop fashion, paving the way for Zero Touch Networking [1]. Optical telemetry is a hot topic in the context of disaggregated optical networks [2, 3], thanks to the availability of open data models allowing the exchange of status information between optical node components belonging to different vendors [4] and centralized controllers and monitor handlers. Such platforms facilitate the introduction of machine learning (ML) engines to process online optical data in the data lake.

With the ultimate goal to enable large and fully automated networks, capable of performing policy-driven self-diagnostics [5, 6], most works leverage the advantages of using supervised learning strategies to a variety of goals, ranging from management to forecasting and event detection [7–9]. Among the plethora of approaches, the ones based on artificial neural networks (ANNs) are undoubtedly the most prominent [10–12].

In this rapid evolving ecosystem, the collection and transfer of the enormous volume of telemetry data brings concerns on the data security, privacy and confidentiality-preservation [13–15]. Particularly for centralized network scenarios, involving the centralization of intelligent data processing modules in a third-party cloud component, preserving the data confi-

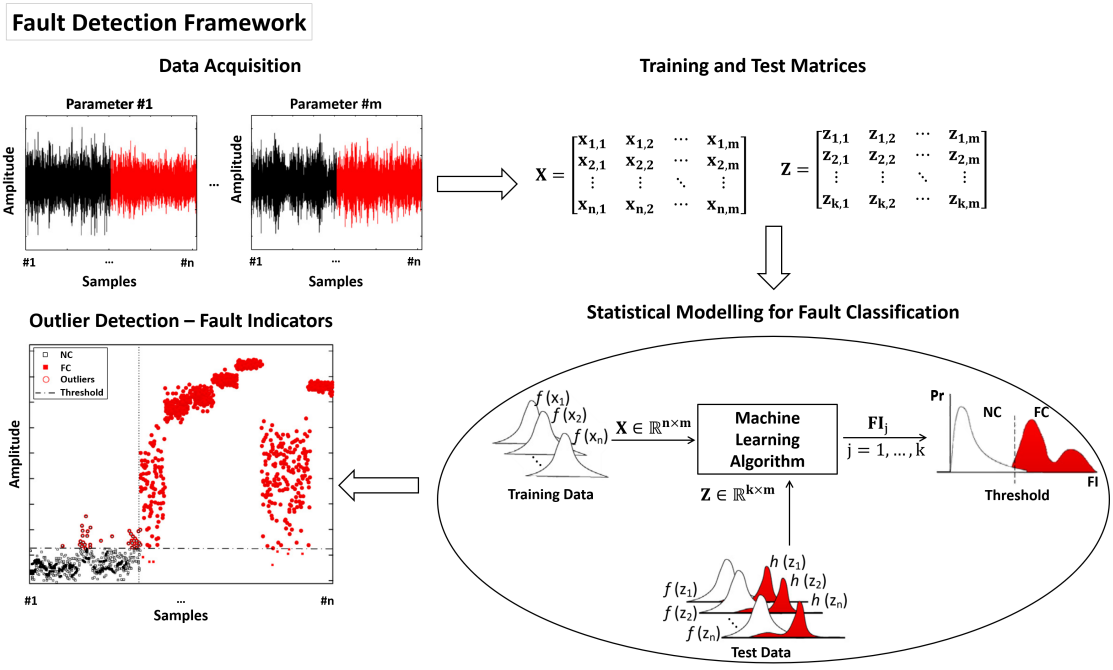

Fig. 1. Overview of the novel detection strategy developed to detect soft failures at the DCO agent.

dentiality could avoid potential information leakage through eavesdropping [16], typically aimed at disrupting the service or gaining unauthorized access to carried data, causing network-wide service disruption and possibly leading to huge data and revenue losses [17, 18]. However, it is important to note that in the context of SDN control, even with the advent of the Elastic Optical Network (EON) disaggregation, the confidentiality of the telemetry information is still at risk as intensive data exchange towards third party telemetry collectors of multi-vendor optical transceivers and pluggable cards are inevitable. Therefore, integrating privacy engineering practices into the early stages of network design arguably leads to better and more secure systems [19].

An emerging approach to ensure data security in the optical network data plane is physical encryption methods, capable to ensure the secrecy of in-flight data in the transport layer [20]. Optical encryption, for instance, exploits the coherent nature of the laser beams, commonly, based on the phase modulation of light using the direct superposition of phase masks containing the original data and an encrypting phase key [20]. Similarly, in the control and management plane, concerns may raise when data are processed by a third-party AI/ML algorithm provider. Indeed, optical network disaggregation may enable third-party telemetry-driven analytics services thanks to open YANG models[21]. In this specific case, network providers might not want to reveal their devices performance to preserve confidentiality. Although this offers clear advantages, additional hardware is required for the physical encryption, increasing budgetary and processing costs.

Alternatively, in this work, we propose a simple yet highly efficient homomorphic solution applied to soft-failure detection in optical communication networks. As demonstrated in [22] and in [23] targeting a different topic related to image processing, a class of machine learning algorithms, rooted in the branch of dimensionality reduction techniques, share a particular property regarding to data rotation axis that can be further exploited to scramble the telemetry data in a fast and effective way, hiding the original content. Based on this invariance property, the current paper is an extension of [24], where the authors previously demonstrated that, the Principal Component Analysis (PCA) algorithm, a classical dimensionality reduction technique, is capable to achieve a very similar performance in soft-failure classification for a specific telemetry dataset from an optical communication system using the data in its original form and in a randomly scrambled version. This allows to transmit telemetry data from an optical system to untrusted third-party cloud computing resources for analysis, without revealing sensitive spatial geometry information contained in the data, reducing security and confidentiality concerns that arise with the deployment of on-cloud processing solutions for network condition assessment and prognostics.

Here, we demonstrate that this concept can be extended to other dimensionality reduction algorithms sharing the same working principals as PCA, notably the ones based on Factor Analysis (FA), Nonlinear Principal Component Analysis (NLPCA), and Singular Value Decomposition (SVD).

A. Related work on ML-driven data security and confidentiality in disaggregated optical networks

Data confidentiality and security is becoming a crucial topic in the context of disaggregated networking. In fact, disaggregation enables open models to exchange control messages and monitoring/telemetry data from/to different devices and white boxes owned by different domains, vendors and operators. Thus, data integrity and sovereignty need to be preserved and assured from external entity attempts or simply to guarantee the privacy policies enforced by service level agreements in multi-vendor environments.

In the context of disaggregated optical networking, recent works are exploring ways to detect, analyze and prevent physical attacks to optical network infrastructures aiming at retrieve traffic data in a malicious way, such as jamming, polarization

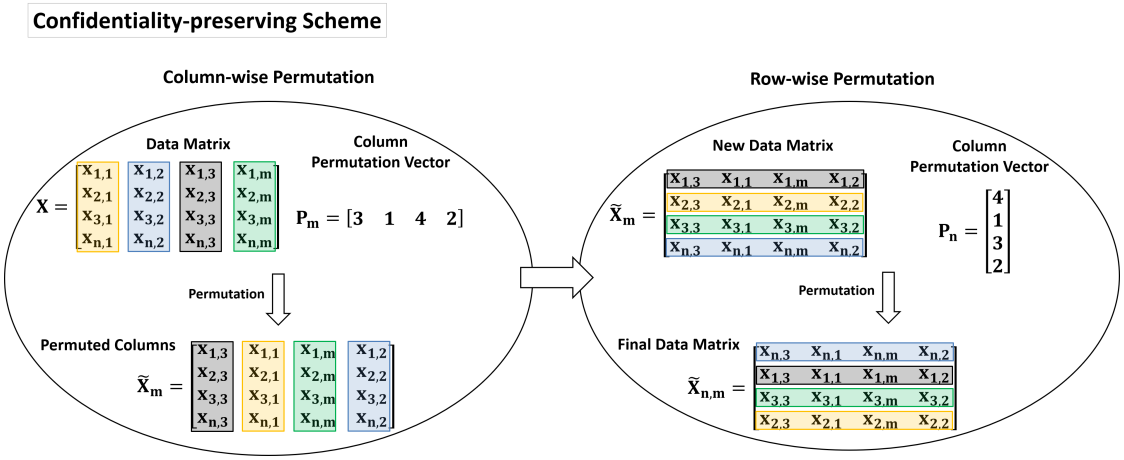

Fig. 2. Example of the confidentiality preservation scheme for soft-failure detection.

modulation attacks, polarization scrambling attacks. The work in [25] surveys the attack techniques at the optical layer and introduces the most considered security management techniques to detect and mitigate attacks, with special focus to machine learning, including supervised, semisupervised and unsupervised techniques, with a detailed analysis on achieved accuracy. The same authors propose in [26] a window-based attack detection approach to address false positives events, also able to perform attack localization at the link level. In [27] proposes the adoption of a root cause analysis to identify attacks at the optical physical layer that can not be detected using machine learning tools or well-established anomaly signatures.

In the context of telemetry data retrieved by optical monitors needed to be consumed by different entities, the work in [28] presents a framework for partially disaggregated networks enabling vendors and operators to develop a common ML-enabled environment without revealing confidential data, resorting to a vertical federated learning solution and a third party training coordinator node. The work in [29] proposes the adoption of a federated marketplace based on the Acumos AI tool to trade a Quality of Transmission (QoT) classifier across different operators while preserving the ML model trained by different operator managers, thanks to a distributed learning framework. The issue of transforming raw data for ML inference to preserve confidentiality have been analyzed in [30] in the use case of image detection, where different transformations have been compared, with the evaluation of the model accuracy on the transformed data and a discussion on the effectiveness and the feasibility of different methods.

Finally, data integrity, auditability and SLA violation responsibility in disaggregated networks have been analyzed in the work in [31], proposing a blockchain logging mechanism enabling full trustworthiness of each device control and management operation with the ability of detecting SLA violations in multi-vendor nodes and systems and univocally identifying device responsibilities due to hardware or software issues.

2. MACHINE LEARNING ALGORITHMS FOR SOFT-FAILURE DETECTION

In real-world applications ML solutions are instrumental to intelligently process the monitoring data of large telemetry systems. These ML-based approaches are, therefore, mandatory for implementing zero-touch optical networks. Amid several possible schemes, here, we introduce a set of unsupervised ML techniques that have the advantage of using only data from normal operation conditions for the model training. Avoiding the collection of failure data for training purposes is advantageous because it eases the data collection process as one does not need to acquire or simulate data from failure conditions.

Thus, for all of the ML techniques here described, data from failure conditions are not used for training. They are implemented on the basis of a binary classification framework (failure or/and no failure) with access to data from normal conditions only during the training phase, being commonly described as unsupervised outlier detection methods.

For the sake of clearness, consider the training data matrix $\mathbf{X} \in \mathbb{R}^{n \times m}$ as composed of m telemetry parameters collected n times from several different network devices under normal working conditions, and the test matrix $\mathbf{Z} \in \mathbb{R}^{k \times m}$ that may be composed of data from the normal condition (NC) and the failure condition (FC). Taking into account the novelty detection strategy summarized in Fig. 1, the training phase is carried out using the collected telemetry data from the optical network. Then, training and test matrices are created, where the training set is composed of samples from the normal working condition of the system only, whereas the test matrix may be composed of data from both normal and failure conditions. Note that in Fig. 1, $f(x_1), ..., f(x_n)$ stands for the distribution of the collected data under normal conditions for the training phase, while $f(z_1), ..., f(z_n)$ and $h(z_1), ..., h(z_n)$ indicate the distribution of the test data collected under normal and failure conditions, respectively. With both matrices at hand a statistical modelling phase is carried out. This step is the actual training phase, where a machine learning model is trained using the training set and tested with the test matrix. At the end of the training step, linear thresholds are estimated for each telemetry parameter based on confidence level over the training data. This step is accomplished after estimating failure indicators for the training data matrix, sorting the resulting values and then selecting as a threshold the value corresponding to the 95% of the sorted data. The end result is that approximately 95% of the failure values from the normal condition will be smaller than the defined threshold, allowing 5% of misclassifications to be taken into account as regular system variations. This approach has been widely adopted in the literature of outlier detection [32, 33]. For completeness, in Fig. 1, at the end of the statistical modelling

for fault classification, one can notice what could be the possible probability distribution (Pr) of the resulting fault indicators for both normal and fault conditions. In this case, the threshold defined over 95% of confidence is marginal and thus allows to the identify the occurrence of failures.

Later, at the failure detection phase, using the trained ML model, the residual error and the failure indicators for the test matrix are calculated. Finally, the samples whose failure indicators surpass the predefined classification threshold are marked as soft failure and can be plotted for visual analysis. The basic intuition is that, if the training has been properly performed, when the mapping/demapping occurs using data from failure conditions in the failure detection stage the residual error grows proportionally to the level of discrepancy between the telemetry from the normal conditions and the failure conditions.

The inputs for the model are simply the raw telemetry parameters collected over time, without any further processing of the input except for a zero-score normalization to alleviate variance problems caused by different scales. The training/testing strategy here described is simple and generic enough to enable its application on single-hop and multi-hop networks by only adding the required telemetry parameters in the input dataset. If possible or required, this approach can also be implemented in a third-party Optical Line Systems (OLS) agent with minimum adaptation.

A. Factor Analysis

Classic factor analysis (FA) is a multivariate technique describing the correlation among a number of observed dependent variables, m, in terms of a linear combination of a small number of unobserved independent variables, $d(d < m)$, also called factors. This linear model in the matrix form is written as:

$$\mathbf{X} = \mathbf{A}\mathbf{E} + \mathbf{e}, \tag{1}$$

where $\mathbf{A}$ is the matrix of factor loadings, $\mathbf{E}$ a matrix of factor scores, and $\mathbf{e}$ the matrix of error, which are assumed to be independent with respect to the specific variances, $\mathbf{\Psi}$. In the context of fault detection, the factor model can be used for data normalization as follows. When estimating the FA model matrix (factor loadings) the normal condition of the system is established using the covariance $\mathbf{\Sigma}$ of the $\mathbf{X}$ under the assumptions of [34]:

$$\mathbf{\Sigma} = \mathbf{A}\mathbf{A}^T + \mathbf{\Psi}, \tag{2}$$

where $\mathbf{\Psi}$ is a diagonal matrix with the corresponding variances. For the test matrix $\mathbf{Z}$, the factor scores $\hat{\mathbf{E}}$ are estimated using,

$$\hat{\mathbf{E}} = \mathbf{A}^T(\mathbf{\Psi} + \mathbf{A}\mathbf{A}^T)^{-1}\mathbf{Z}. \tag{3}$$

This compressed version of $\mathbf{Z}$ being $\hat{\mathbf{E}} \in \mathbb{R}^{k\times d}$, is then reconstructed and used to compute the fault indicators on the basis of the Euclidean norm as:

$$\mathbf{FI} = \|\mathbf{Z} - \mathbf{A}\hat{\mathbf{E}}\|. \tag{4}$$

B. Principal Component Analysis

Principal Component Analysis (PCA) is a classical multivariate statistical procedure that estimate a linear static relationship between the input data and a small unknown number of latent variables that retain most of the variance in the data [35]. This subspace mapping is reached by reducing the dimensionality of the original input data through a linear projection. The training matrix, $\mathbf{X}$, is decomposed into:

$$\mathbf{X} = \mathbf{T}\mathbf{U}^T, \tag{5}$$

where $\mathbf{T}$ is the scores matrix and $\mathbf{U}$ a set of m orthogonal vectors (loadings matrix). These vectors are obtained by decomposing the covariance matrix $\mathbf{\Sigma}$ of $\mathbf{X}$ using:

$$\mathbf{\Sigma} = \mathbf{U}\Lambda\mathbf{U}^T, \tag{6}$$

where Λ is a diagonal matrix containing the ranked eigenvalues, and $\mathbf{U}$ is the matrix containing the corresponding eigenvectors. The first d eigenvectors associated with the higher eigenvalues are the principal components as they correspond to the dimensions that have the largest variability in the data.

This allows to perform an orthogonal transformation of the training data matrix $\mathbf{X}$ by retaining only the principal components d ($\leq m$). Choosing only the first d eigenvectors, the final matrix can be rewritten without significant loss of information in the form of:

$$\hat{\mathbf{X}} = \mathbf{T}_d\mathbf{U}_d^T + \mathbf{E}, \tag{7}$$

whose dimensions are $\mathbf{T}_d \in \mathbb{R}^{n\times d}$ and $\mathbf{U}_d \in \mathbb{R}^{m\times d}$, and therefore reconstructing the original input data matrix. In this case, $\mathbf{E}$, is the residual matrix resulting from the d factors. The coefficients of the linear transformation are such that if the feature transformation is applied to the training data set and then reversed, there will be a negligible reconstruction error.

With the principal components at hand, to detect failures from unseen data, a mapping/demapping operation using the main orthogonal vectors $\mathbf{U}_d$ is required. Assuming the test matrix, $\mathbf{Z}$, that may contains data from both normal and failure conditions, the residual error, $\mathbf{E}$, is computed as the difference between the original and the reconstructed test matrix using:

$$\mathbf{E} = \mathbf{Z} - (\mathbf{Z}\mathbf{U}_d)\,\mathbf{U}_d^T. \tag{8}$$

Later, to track possible failures based on this residual error, the Euclidean norm is performed over each value of $\mathbf{E}$, thus resulting in $\mathbf{FI}$.

C. Nonlinear Principal Component Analysis

A common approach to implement a nonlinear principal component analysis (NLPCA) is using autoassociative neural networks (AANN). Introduced by Kramer [36], the AANN is trained to characterize the underlying dependency of the identified features on an unobserved small set of hidden factors by treating that unobserved dependency as hidden intrinsic variables in the network architecture. More precisely, the ANN architecture consists of three hidden layers: the mapping layer, the bottleneck layer, and demapping layer. More details on the network, including the number of nodes to use, can be found in the references [34, 37].

This algorithm can be thought as a semi-supervised learning approach, as it is a mixture of two different learning schemes, i.e., supervised learning is used to obtain the independent hidden factors, while unsupervised learning is used to perform failure detection as for the training phase only data from normal condition is used while in the test phase one may have data from both normal and failure conditions.

In our context, the AANN is first trained to learn the correlations between features from the training matrix, $\mathbf{X}$, composed of data from the normal condition. The trained network should be able to quantify the hidden unobserved factors that nonlinearly compose the original data. This small set of hidden variables

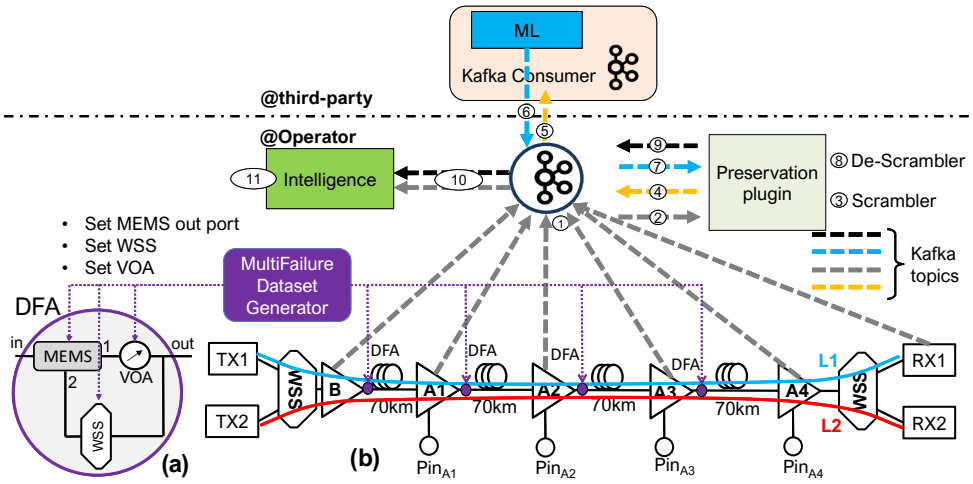

Fig. 3. a) Distributed Failure Actuator (DFA) internal architecture and configurability; b) Experimental optical testbed and telemetry system.

is represented at the bottleneck output of the model, where the number of nodes (or factors) should correspond to the number of unobserved independent factors.

For the test matrix $\mathbf{Z}$, the fault indicators $\mathbf{FI}$ is given by:

$$\mathbf{FI} = \|\mathbf{Z} - \hat{\mathbf{Z}}\|. \tag{9}$$

Similarly to the expected behavior for the FI values, if the l feature vector is related to a normal condition its $\mathbf{FI}_l \approx 0$. On the other hand, if the feature vector comes from the failure condition, the AANN should not be able to predict the targets, and then, the corresponding $\mathbf{FI}_l$ values deviates from zero, increasing proportionally to the severity of the failure.

D. Singular Value Decomposition

The singular value decomposition (SVD) algorithm aims at determining the rank of a state matrix $\mathbf{M}$. For the training matrix $\mathbf{X} \in \mathbb{R}^{n\times m}$, a possible alternative to compose this state matrix is by fixing the training matrix and, for each incoming new feature vector from a test matrix $\mathbf{Z}_l \in \mathbb{R}^{1\times m}$, define:

$$\mathbf{M}_l = [\mathbf{X}, \mathbf{Z}_l] \in \mathbb{R}^{(n+1)\times m}. \tag{10}$$

For the training stage, the SVD factorization, considering only real values, is defined as:

$$\mathbf{M} = \mathbf{UAV}, \tag{11}$$

where $\mathbf{U}$ and $\mathbf{V}$ are two orthogonal matrices and $\mathbf{A}$ contains the singular values of matrix $\mathbf{M}$ along its diagonal line sorted in descending order. To discriminate the normal and failure conditions, one must first estimate the singular values, $\mathbf{A}_{train}$, using $\mathbf{M} = \mathbf{X}$. Then, for each feature vector l from the test matrix, the singular values are computed by concatenating the feature vector l with the training matrix using Equation 10. The corresponding singular values are stored in the diagonal matrix $\mathbf{A}_l$. After diagonalizing $\mathbf{A}_{train}$ and $\mathbf{A}_l$ as $\theta_{train} = [\theta^1_{train}, \ldots, \theta^m_{train}]$ and $\theta_l = [\theta^1_l, \ldots, \theta^m_l]$, respectively, the fault indicator is calculated by using:

$$\mathbf{FI}(l) = \|\theta_{train} - \theta_l\|. \tag{12}$$

3. CONFIDENTIALITY-PRESERVING SCHEME

A relevant aspect of the the four machine learning techniques described before, is their mapping/demaping operations. Besides each particular projection mechanism, these techniques aim at learning a manifold space containing hidden factors independent from each other. Relying on a key property, we observed that, no mattering the ordering of the measurement instances, the rank of a given training matrix does not change along with different types of data permutation one can assume of the data. This results in no effect on the resulting calculation of the latent variables using the different decomposition techniques.

For instance, a training matrix $\mathbf{X} \in \mathbb{R}^{n\times m}$ can be rearranged without any lack of performance of the ML algorithms by randomly swapping the rows and the collumns of the matrix, resulting in the permuted version of the training matrix $\hat{\mathbf{X}} \in \mathbb{R}^{n\times m}$. This can be thought as a rotation invariant property, which can be exploited to achieve confidentiality preservation by simply shuffling the telemetry data.

Finding the correct spatial ordering of a set of collected measurements is analogous to trying to correctly reassemble a common jigsaw puzzle. The work in [38] showed that this problem falls into the NP-complete class, and is therefore an unfeasible problem to be optimally solved computationally, with no efficient algorithm available up to this date. Thus, the scrambling order acts as a secret key that allows proper reconstruction of a dataset. In our case, the telemetry data is composed of several time-series with time-dependant structure. For a large dataset, if the time order is changed, the data becomes unintelligible for a potentially unauthorized third-party observer.

For a clear illustration, a simple scrambling operation over the matrix $\mathbf{X}$ could involve the random swapping of its rows using a function $[\tilde{\mathbf{X}}_n, \mathrm{P}_n] = perm(\mathbf{X}, n)$, where n indicates a random swapping of entire rows of the given matrix. A different but also effective operation could be the random swapping of columns using $[\tilde{\mathbf{X}}_m, \mathrm{P}_m] = perm(\mathbf{X}, m)$, where m denotes the random swapping of entire columns. Apart from returning a scrambled version of the input dataset, this swapping function also keeps a vector P with the correct order of the data in the training matrix that works as an re-ordering key holded by the data owner or manager. Whenever necessary this unscrambling vector can be used to rearrange the matrix to its original form.

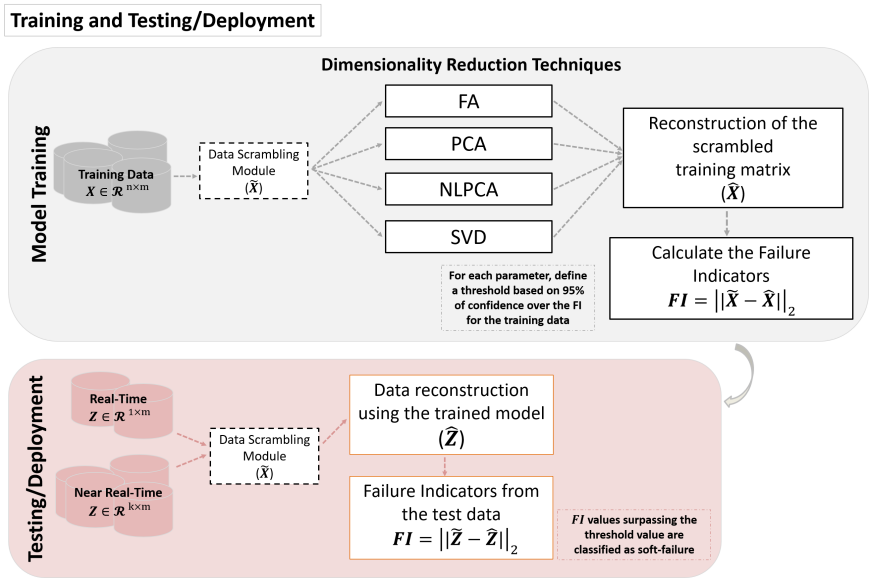

Fig. 4. Overview of the novelty detection strategy to detect soft failures in the context of the proposed confidentiality-preserving scheme.

Applying any of the machine learning techniques over $\tilde{\mathbf{X}}_n$ or $\tilde{\mathbf{X}}_m$ results in approximately the same manifold space, but with different rotations. This further implies that a deployed monitoring system can send spatially-scrambled data to a third-party cloud service to perform failure detection without any concern for the spatial information associated with the structure being revealed. A similar result holds for shuffling the data using both approaches, whose difference still in the rotation of the learned manifold space. An example of this framework is shown in Figure 2. In this case, column permutation followed by a row permutation is applied to create a two-layer permuted data matrix $\tilde{\mathbf{X}}_{n,m}$.

Therefore, from the described principle of rotation invariance, we can derive a general framework to detect soft-failures with privacy-preserved telemetry data. After the data collection, we can choose one of the scrambling approaches to the training dataset (here, we choose to use both operations, at same time). This step results in a random key that later is used to unscramble the results. The scrambled dataset is then send to a third-party cloud service where the ML algorithm can be applied in a straightforward manner, returning to the client only the matrix with the unobserved latent factors. Next, failure detection is performed as described for each algorithm. At this stage the random key is used to unscramble the results before display.

Please note that after deployment of the trained model a few operation options are available in terms of scalability. The network manager can chose to implement this solution in a real-time or near real-time fashion depending on the available resources and system requirements, as is illustrated in Figure 4. The real-time operation includes a simple scramble of single measurements before sending it to the third-party cloud partner. In other words, the system could be set to simply perform random swapping of the columns in $[\tilde{\mathbf{X}}_m, \mathrm{P}_m] = perm\,(\mathbf{X}, m)$, where $\mathbf{X} \in \mathbb{R}^{1 \times m}$ would be a simple line vector, involving a neglectable processing time and network throughput. Alternatively, the manager can also chose to implement the solution as near real-time by creating a buffer of measurements and therefore being capable to perform both row and column scrambling over the collected data. Please also note that from the point of view of the proposed technique the failure detection performance is the same regardless of the chosen implementation.

4. EXPERIMENTAL EVALUATION SCENARIO

The proposed machine learning algorithms have been experimentally evaluated in a optical network testbed able to reproduce configurable soft failure events, shown in the bottom of Fig. 3. In the top part of the figure, the data collection architecture is shown, highlighting the functional blocks and the steps of the considered workflow. The testbed employs a physical link composed by four 70 km-long single mode fiber spans. At the end of each span an EDFA amplifier A_i is placed, configured in gain mode at around 15dB in order to compensate the optical signal loss along the span. The optical signals (100G, 37.5GHz spectrum width, PM-QPSK modulation format) are generated by two commercially available 100G muxponders. The two resulting lightpaths, i.e., $L1$ and $L2$, are multiplexed using a Wavelength Selective Switch (WSS) and then amplified by a booster amplifier B in power mode, such that each single channel is launched with 0dBm optical power in the first span in order to avoid detrimental non-linear effects along the link. After the four spans, demultiplexing is realized through an additional WSS, thus each signal is sent to a different coherent receiver equipped with the Digital Signal Processing (DSP) providing channel performance indicators (i.e., estimated OSNR and pre-FEC BER).

To emulate soft failures, a specific optical device called Distributed Failure Actuator (DFA) has been assembled and placed at each span. The device internal scheme is shown in Fig. 3(b). The input port is sent to a Micro Electro-Mechanical Switch (MEMS) with two alternative outputs. The former output is sent to a Variable Optical Attenuator (VOA), emulating incremental line loss, while the latter implements narrow filtering (i.e., attenuation on a single selected channel) using a WSS. The two output lines are sent to a single DFA output by means of a coupler. The VOA is configured in order to achieve the same attenuation for the two DFA lines, considering the MEMS (i.e.,

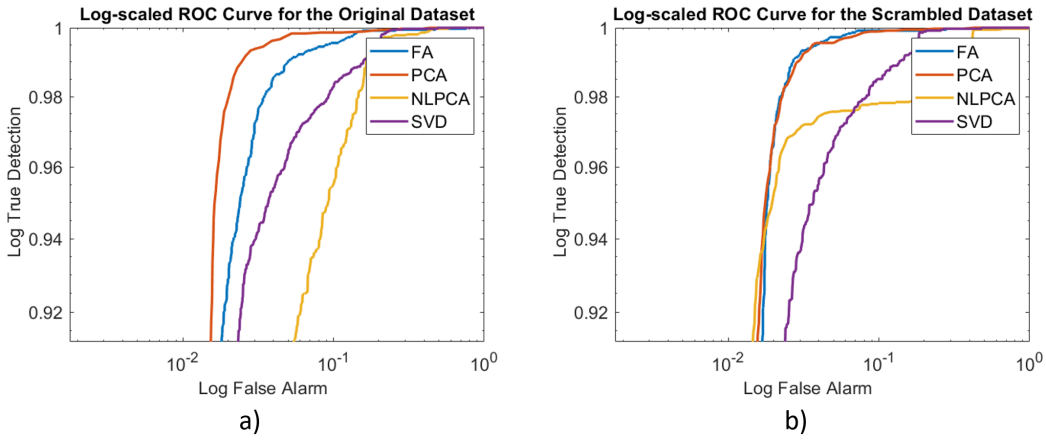

Fig. 5. Log-scaled ROC curves for the FA, PCA, NLPCA and SVD algorithms: a) original and b) scrambled datasets.

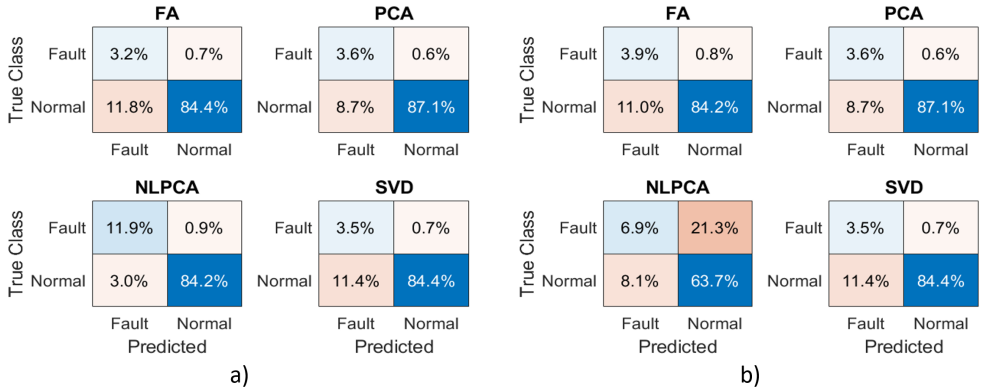

Fig. 6. Confusion matrix for the FA, PCA, NLPCA and SVD algorithms: a) original and b) scrambled datasets.

3dB) and the WSS (i.e., 8dB) insertion loss contributions. This way, by dynamically configuring the MEMS output port and the WSS filter, it is possible to emulate three possible types of soft failure at each span: a) a generalized 10dB attenuation over the two channels (MEMS output port 2 and WSS filtering both $L1$ and $L2$); b) a 10dB degradation of $L1$ only (MEMS output port 2 and WSS $L1$ filter); a 10dB degradation of $L2$ only (MEMS output port 2 and WSS $L2$ filter).

A custom monitoring platform enhanced with the new-generation telemetry procedures described in [2] has been adopted to collect the main performance parameters from the considered optical network.

The platform, based on *Kafka*, implements a messages streaming system, allowing the transmission of performance parameters. This solution avoids the uage of specific telemetry protocols, but allows the streaming of JSON-based messages, conveying the metric in a (key, value) format. At step 1, the metrics are collected from the optical devices, enhanced with a *KafkaProducer*, and are transmitted, over specific *Topics*, towards a data collector, equipped with a *KafkaConsumers*, subscribed to the topics. In example, the *OpticalNetowork* topic has been defined to store all the metrics collected form the underlying devices. A consumer, subscribed to the *OpticalNetwork* topic, will receive all the data from the devices. Considering the reference scenario, shown in Fig. 3, the parameters collected from the devices include: input power level for each amplifier (e.g., Pin_{Ai} in the figure, with i=1,2,3,4) and the Quality of Transmission (QoT) parameters for each active lightpath, estimated by the DSP at receiver side. The considered coherent data are the OSNR and the instantaneous pre-FEC Bit Error rate. All the telemetry metrics are collected with a rate of 5 seconds.

The monitoring architecture allows to process the data in real-time, using dedicated processing nodes, based on Apache Streams, in order to consume, process (i.e., augment, filter, scramble or scale) and produce new data. The processing nodes can be plugged to the Kafka Brokers in the form of plugin.

In the proposed solution, the processing nodes have been adopted to implement the core functionality to obtain the data confidentiality towards the third-party app.

In particular, the designed plugin, after the reception of the telemetry data, on topic *OpticalNetwork*, represent with a grey arrow in the figure, from the devices (step 2), performs the data scrambling (step 3 in the figure), by executing the data features permutation. Then, the new data is transmitted over a dedicated topic, i.e., *Confidentiality* (represented with a yellow arrow at step 4). The scrambled data is received (step 5) by the third-party entity, subscribed to the *Confidentiality* topic. The ML algorithm is executed and the results are transmitted on the *MLResults* topic (step 6). The Preservation plugin, at the operator side, is subscribed also to the *Confidentiality* topic, receiving the data to be de-scrambled (steps 7-8). The re-ordered data is then transmitted on the *Validation* topic, represented with a black arrow at step 9, being transmitted to the Intelligence module, (step 10), subscribed to both the *OpticalNetwork* and the *Validation* topics, being able to compare the ML-based reconstructed data with the real one, checking for the presence of

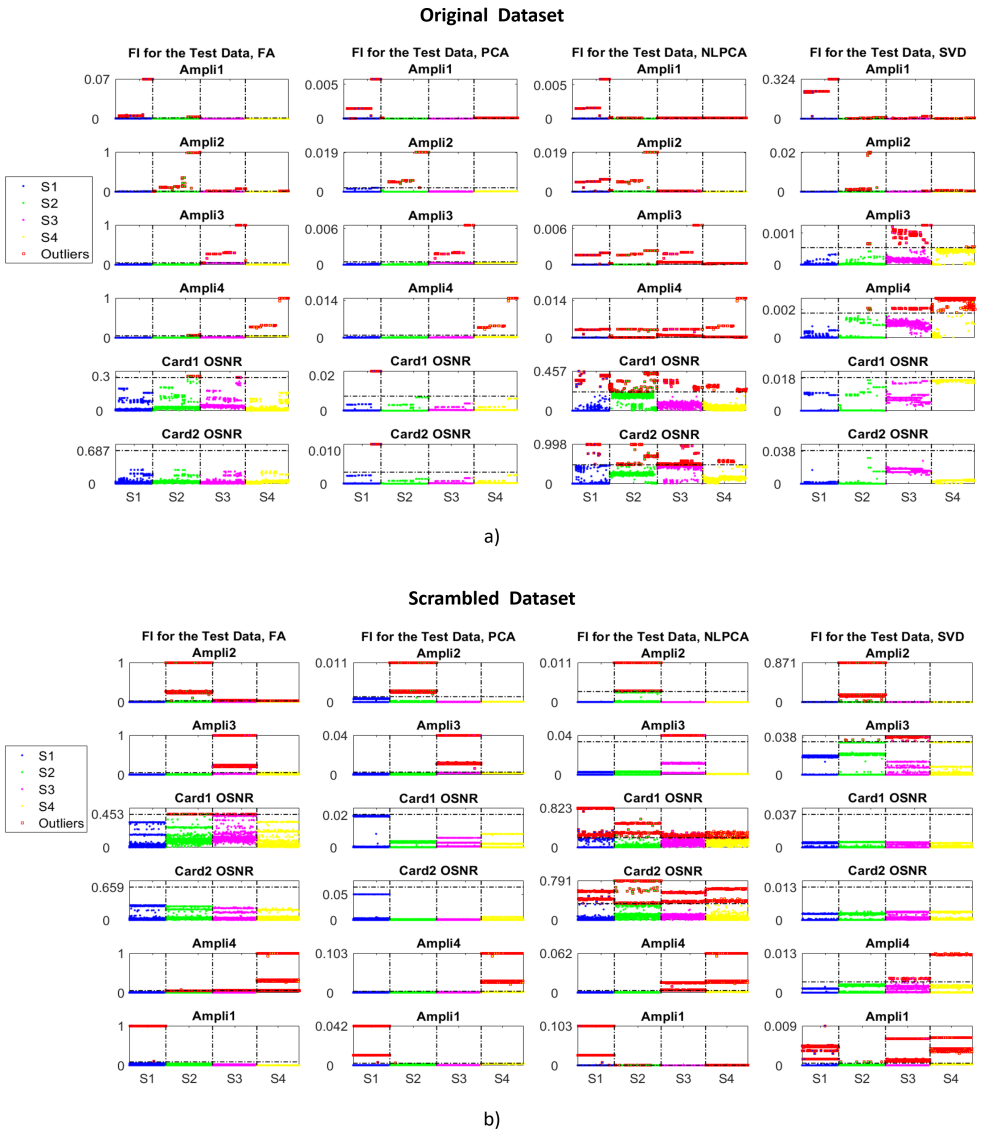

Fig. 7. Outlier detection for the FA, PCA, NLPCA and SVD algorithms: a) original and b) scrambled datasets. The scrambled dataset has the variables in the order of a random permutation performed over the dataset.

outliers (step 11), related to failures occurring in the system.

The described monitoring system has been used to collect 5 different datasets: a 24h long dataset, for training, and 4 1h long datasets for testing. Each dataset includes 6 metrics collected from the testbed: the input power at each in line amplifier and the pre-fec-BER extimated at the receivers of the two lightpaths. The training dataset considers only normal operation with regular fluctuation of the different metrics due by the variation of side conditions. Each testing dataset presents 15 minutes of normal behavior and then considers a sequence of failures in one of the 4 spans (i.e., dataset1 considers failures only on span1 (S1), dataset2 failures only on span2 (S2), dataset3 failures only on span3 (S3) and dataset4 failures only on span4 (S4)). In particular, each failure dataset, after the collection of metrics under normal conditions, includes the sequence of failures composed of three phases: (I) five failures on $L1$ only, (II) five failures on $L2$ only, (III) failures on both $L1$ and $L2$. The inter-failure time is randomly selected in the range 1 and 4 minutes and each failure has a fixed duration of 1 minute.

5. EVALUATION AND RESULTS

For generalization purposes, the optical dataset was split into training and test matrices. The training matrix, $\mathbf{X} \in \mathbb{R}^{16468\times 6}$, permits each algorithm to learn the underlying distribution and dependency of all variability states of the optical system, considering only normal working condition. The test matrix, $\mathbf{Z} \in \mathbb{R}^{7737\times 6}$, includes the state parameters of the optical system in both normal and failure conditions. In this case, the 20% of the entire data related to the normal condition was used for validation purposes only. The remaining data, containing samples from failure condition, is also included in the testing. This procedure permits to evaluate the generalization performance of the considered machine learning algorithms in an exclusive manner, because the time-series metrics, used in the test phase, are not included in the training phase. During the test phase, the algorithms are expected to detect deviations from the normal condition when the data vectors come from failure condition in the optical system. The next step is to carry out statistical modeling for feature classification.

In that regard, the algorithms based on FA, PCA, NLPCA and SVD are implemented in an unsupervised learning mode by first taking into account features from all the normal state conditions (training matrix). The FA algorithm was implemented on the basis of the classical Maximum Likelihood common Factor Analysis [39]. In counterpart, the PCA, NLPCA and SVD were implemented the procedures described in [37]. For all the ML algorithms, 3 hidden factors were extracted for the considered optical dataset. Note that, for the NLPCA, the number of hidden factors is equal to the number of nodes on its bottleneck layer.

The Receiver Operating Characteristic (ROC) curves provide a comprehensive means of summarizing the performance of classifiers. 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. Figures 5a and 5b plot the ROC curves for the ML-based algorithms in log-scale considering the two cases, when the optical dataset is kept in its original form and for its scrambled version, to ease the visualization. Qualitatively, looking at the curves, for both cases, one can verify that none of the algorithms can have a perfect classification with a linear threshold, because none of the curves go through the left-upper corner, neither have supremacy in terms of true detection rate for all the false alarm domain. Furthermore, one can verify that for levels of significance around 5 percent, the FA and PCA have better true detection rate than NLPCA and SVD, i.e., the approaches that maximize the true detection of failure cases with similar performances in terms of false alarm rate. Nonetheless, for low probabilities of false alarm, FA and PCA still demonstrate an acceptable true detection rate (for instance, for a false alarm rate of 0.05 for both algorithms, the minimal true detection rate is around 0.98).

Note that all proposed algorithms apply data transformation in the original feature space to achieve a data model that represents the normal system condition, thus sharing similar working structure but relevant differences in terms of classification performance. For instance, the NLPCA has quite different classification performance for the original and the scrambled datasets, as well as the FA algorithm. This is due to both techniques use heuristic procedures during its learning phase, resulting in slightly different performance for each run. Only PCA and SVD maintain the exact performance for both datasets, as they make use of determinative optimization algorithms. Specifically for PCA and SVD, this is relevant to note, as it clearly demonstrates that, if the learning algorithm is not subjected to random factors, the expected classification performance of the ML technique should be exactly the same, being transparent to any permutation performed over the dataset.

In that regard, to clearly demonstrate that, for deterministic ML techniques, the classification performance is not affected by the permutation on the dataset, Figure 6 presents the confusion matrices for both original and scrambled datasets. In general, PCA reaches the best classification performance for both cases, with an accuracy rate over 90%. FA and SVD algorithms also perform well on both datasets, with quite similar performance when classifying samples from normal conditions (greater than 87.6%). Only the NLPCA demonstrates quite different classification performance, with very poor true classification rate when working over the scrambled dataset. These results agree with what is observed for the ROC curves, showing that PCA and SVD, because of their deterministic optimization schemes, are capable to maintain the exact level of performance for the different versions of the optical dataset, where PCA demonstrates the best classification accuracy.

In order to quantify the performance of the classifiers for a given threshold, Figures 7a and 7b plot the fault indicators (FIs) for the feature vectors of the entire test data along with a threshold defined based on the 95 percent cut-off value over the training data. The evolution of the outlier detection is presented considering the failures occurring on one of the spans of the link. With S1, S2, S3, S4 we label the span where the failures occur, highlighting all the metrics collected from the testbed, in order to show the accuracy of the different algorithms and how each failure affects the different metrics in the testbed. All the plots are scaled to the threshold level, in order to highlight the effectiveness of the algorithms in terms of outliers detection. For the original dataset (Figure 7a), when evaluating each individual telemetry parameter, only PCA was capable to accurately match the failure indications related to all the four spans of the link. The other techniques demonstrate capabilities to indicate the location of true failures (true alarm) but also indicate a substantial number of failures for portions of the data one would not expect (false alarm), indicating that only PCA maintains an adequate trade-off between true detection and false alarm rates, followed by the SVD with a similar classification accuracy. Considering real-world monitoring scenarios, the PCA algorithm shows the best general behavior in terms of failure classification, working directly over the raw telemetry parameters collected over time, without any further processing of the input. The outlier detection strategy here described is simple and generic enough to enable its application on single-hop, multi-hop, many-hop networks by adding the required telemetry parameters in the input dataset. If possible or required, this approach can also be implemented in a third-party OLS agent with minimum adaptation, maintaining the secrecy of the collected data from unauthorized accesses. The entire framework can be easily expanded by card vendors as they may utilize different datasets and models related to different OLS normal conditions, thus the operators can configure the ML module with the specific models provided by the vendor card.

6. CONCLUSION

This paper proposed a simple and effective approach to ensure confidentiality of telemetry data in the context of machine learning-based soft-failure detection in optical communication networks. This confidentiality procedure can be implemented as a protocol agnostic solution which allows the configuration of several other protocols on the top of it running out-of-band without system overhead.

Experimental results showed that this confidentiality procedure can be coupled with different machine learning techniques rooted on the field of dimensionality reduction methods, allowing flexibility when building a solution for network condition assessment. We also demonstrated that a key property of this solution is to allow the run the machine learning solution on a scrambled version of the telemetry data without lacking failure detection performance. Finally, implemented in an unsupervised fashion, the ML algorithms here introduced disregards the need for data from failure conditions of the network system, reducing the computational loads during training and deployment of the proposed solution.

Future works aim at demonstrating robust mathematical proof of security for the proposed confidentiality scheme, along with field implementation and deployment on real-world optical network systems.

ACKNOWLEDGMENTS

This work received funding from the ECSEL JU project BRAINE (grant agreement No 876967). The JU receives support from the EU Horizon 2020 research and innovation programme and the Italian Ministry of University and Research (MUR). The work is also funded by the EU H2020 project B5G-OPEN (grant agreement n. 101016663).

REFERENCES

1. ETSI, "Zero-touch network and service management (ZSM); reference architecture," ETSI GS ZSM 002 V1.1.1 (2019-08) (2019).
2. A. Sgambelluri, A. Pacini, F. Paolucci, P. Castoldi, and L. Valcarenghi, "Reliable and scalable kafka-based framework for optical network telemetry," IEEE/OSA J. Opt. Commun. Netw. **13**, E42–E52 (2021).
3. F. Paolucci, A. Sgambelluri, M. Felipe Silva, A. Pacini, P. Castoldi, L. Valcarenghi, and F. Cugini, "Peer-to-peer disaggregated telemetry for autonomic machine-learning-driven transceiver operation," J. Opt. Commun. Netw. **14**, 606–620 (2022).
4. L. Velasco, A. Sgambelluri, R. Casellas, L. Gifre, J. Izquierdo-Zaragoza, F. Fresi, F. Paolucci, R. Martínez, and E. Riccardi, "Building autonomic optical whitebox-based networks," J. Light. Technol. **36**, 3097–3104 (2018).
5. E. Coronado, S. N. Khan, and R. Riggio, "5g-empower: A software-defined networking platform for 5g radio access networks," IEEE Transactions on Netw. Serv. Manag. **16**, 715–728 (2019).
6. H. Kabir, M. H. Bin Mohsin, and R. Kantola, "Implementing a security policy management for 5g customer edge nodes," in *NOMS 2020 - 2020 IEEE/IFIP Network Operations and Management Symposium,* (2020), pp. 1–8.
7. 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 *2020 Photonics North (PN),* (2020), pp. 1–1.
8. S. Shahkarami, F. Musumeci, F. Cugini, and M. Tornatore, "Machine-learning-based soft-failure detection and identification in optical networks," in *2018 Optical Fiber Communications Conference and Exposition (OFC),* (2018), pp. 1–3.
9. A. Mohamed, H. Ruan, M. H. H. Abdelwahab, B. Dorneanu, P. Xiao, H. Arellano-Garcia, Y. Gao, and R. Tafazolli, "An inter-disciplinary modelling approach in industrial 5g/6g and machine learning era," in *2020 IEEE International Conference on Communications Workshops (ICC Workshops),* (2020), pp. 1–6.
10. 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 *2020 Photonics North (PN),* (2020), pp. 1–1.
11. K. Mayer, J. Soares, R. Pinto, C. Rothenberg, D. Arantes, and D. Mello, "Machine-learning-based soft-failure localization with partial software-defined networking telemetry," J. Opt. Commun. Netw. **13**, E122–E131 (2021).
12. M. F. Silva, A. Pacini, A. Sgambelluri, and L. Valcarenghi, "Learning long- and short-term temporal patterns for ml-driven fault management in optical communication networks," IEEE Transactions on Netw. Serv. Manag. **19**, 2195–2206 (2022).
13. S. Rothe, N. Koukourakis, H. Radner, A. Lonnstrom, E. Jorswieck, and J. Czarske, "Physical layer security in multimode fiber optical networks," Sci. Reports **10** (2020).
14. F. Chen, M. Song, F. Zhou, and Z. Zhu, "Security-aware planning of packet-over-optical networks in consideration of otn encryption," IEEE Transactions on Netw. Serv. Manag. **18**, 3209–3220 (2021).
15. M. L. F. Abbade, P. P. P. Junior, I. E. L. Rodrigues, W. S. Souza, L. H. Bonani, and I. Aldaya, "Security in optical communication systems: Data encryption and beyond," in *2021 SBFoton International Optics and Photonics Conference (SBFoton IOPC),* (2021), pp. 1–6.
16. R. Rejeb, M. S. Leeson, and R. J. Green, "Fault and attack management in all-optical networks," IEEE Commun. Mag. **44**, 79–86 (2006).
17. M. Furdek, N. Skorin-Kapov, S. Zsigmond, and L. Wosinska, "Vulnerabilities and security issues in optical networks," (2014), pp. 1–4.
18. H. Hao, Z. Pang, G. Wang, and B. Wang, "Indoor optical fiber eavesdropping approach and its avoidance," Opt. Express **30**, 36774 (2022).
19. D. Mascareñas, A. Green, M. Silva, and B. Martinez, "Privacy-preserving structural dynamics," in *Data Science in Engineering, Volume 9,* R. Madarshahian and F. Hemez, eds. (Springer International Publishing, Cham, 2022), pp. 237–240.
20. *Optical Encryption and Decryption* (Springer Netherlands, Dordrecht, 2009), pp. 273–298.
21. A. Sgambelluri, J. . Izquierdo-Zaragoza, A. Giorgetti, L. Gifre, L. Velasco, F. Paolucci, N. Sambo, F. Fresi, P. Castoldi, A. C. Piat, R. Morro, E. Riccardi, A. D'Errico, and F. Cugini, "Fully disaggregated ROADM white box with NETCONF/YANG control, telemetry, and machine learning-based monitoring," in *2018 Optical Fiber Communications Conference and Exposition (OFC),* (2018), pp. 1–3.
22. B. Martinez, A. Green, M. F. Silva, Y. Yang, and D. Mascarenas, "Sparse and random sampling techniques for high-resolution, full-field, bss-based structural dynamics identification from video," Sensors **20** (2020).
23. B. Bai, Y. Luo, T. Gan, J. Hu, Y. Li, Y. Zhao, D. Mengu, M. Jarrahi, and A. Ozcan, "To image, or not to image: class-specific diffractive cameras with all-optical erasure of undesired objects," eLight **2**, 1–20 (2022).
24. M. Felipe Silva, A. Pacini, A. Sgambelluri, F. Paolucci, and L. Valcarenghi, "Confidentiality-preserving machine learning scheme to detect soft-failures in optical communication networks," in *ECOC 2022,* (2022), pp. 1–4.
25. M. Furdek and C. Natalino, "Chapter ten - machine learning for network security management, attacks, and intrusions detection," in *Machine Learning for Future Fiber-Optic Communication Systems,* A. P. T. Lau and F. N. Khan, eds. (Academic Press, 2022), pp. 317–336.
26. M. Furdek, C. Natalino, F. Lipp, D. Hock, A. D. Giglio, and M. Schiano, "Machine learning for optical network security monitoring: A practical perspective," J. Light. Technol. **38**, 2860–2871 (2020).
27. C. Natalino, M. Schiano, A. D. Giglio, and M. Furdek, "Root cause analysis for autonomous optical network security management," IEEE Transactions on Netw. Serv. Manag. **19**, 2702–2713 (2022).
28. N. Hashemi, P. Safari, B. Shariati, and J. K. Fischer, "Vertical federated learning for privacy-preserving ml model development in partially disaggregated networks," in *2021 European Conference on Optical Communication (ECOC),* (2021), pp. 1–4.
29. B. Shariati, P. Safari, G. Bergk, F. I. Oertel, and J. Karl Fischer, "Interoperator machine learning model trading over acumos ai federated marketplace," in *2021 Optical Fiber Communications Conference and Exhibition (OFC),* (2021), pp. 1–3.
30. V. Prodomo, R. Gonzalez, and M. Gramaglia, "Trading accuracy for privacy in machine learning tasks: An empirical analysis," in *2021 IEEE Conference on Communications and Network Security (CNS),* (2021), pp. 1–2.
31. S. Fichera, A. Sgambelluri, F. Paolucci, A. Giorgetti, N. Sambo, P. Castoldi, and F. Cugini, "Blockchain-anchored disaggregated optical networks," J. Light. Technol. **39**, 6357–6365 (2021).
32. A. Indurkhya, J. C. Gardiner, and Z. Luo, "The effect of outliers on confidence interval procedures for cost-effectiveness ratios," Stat. Medicine **20** (2001).
33. C. Farrar and K. Worden, *Structural Health Monitoring: A Machine Learning Perspective* (2013).
34. E. Figueiredo, G. Park, C. R. Farrar, K. Worden, and J. Figueiras, "Machine learning algorithms for damage detection under operational and environmental variability," Struct. Heal. Monit. **10**, 559–572 (2011).
35. I. Jolliffe, *Principal Component Analysis* (2nd Edition. Springer, 2002).
36. M. A. Kramer, "Nonlinear principal component analysis using autoassociative neural networks," AIChE J. **37**, 233–243 (1991).
37. E. Figueiredo, G. Park, J. Figueiras, C. Farrar, and K. Worden, "Structural health monitoring algorithm comparisons using standard data sets," (2009).
38. E. D. Demaine and M. L. Demaine, "Jigsaw puzzles, edge matching, and polyomino packing: Connections and complexity," Graph. Comb. **23**, 195–208 (2007).
39. S. Sharma, *Applied Multivariate Techniques* (John Wiley & Sons, Inc., USA, 1996).