Confidentiality-Preserving Real-Time Localization of Soft Failures in Optical Networks based on PCA and MLaaS

AZARM YEGANEHFALLAH[1,*], ANDREA SGAMBELLURI[1], EMILIO PAOLINI[1], KAYOL SOARES MAYER[2], MOISES FELIPE SILVA[3], DARLI A. A. MELLO[2], AND LUCA VALCARENGHI[1]

[1] *Scuola Superiore Sant'Anna, Pisa, Italy*
[2] *School of Electrical and Computer Engineering, Unicamp, SP, Brazil*
[3] *Los Alamos National Laboratory, USA*
[*] *azarm.yeganehfallah@santannapisa.it*

Abstract:

Proactive management of soft failures is crucial for enhancing the reliability of optical networks. However, developing solutions that are simultaneously accurate, operate in real-time, ensure data confidentiality, and scale effectively represents a significant challenges.

This paper proposes a method for soft failure localization that ensures data confidentiality. The approach is devised for a scenario where the data owner (e.g., the network provider) elaborates its confidential data (e.g., telemetry data) through machine learning services provided by a third party (i.e., Machine Learning as a Service — MLaaS). Data confidentiality and, as an important by-product, reduced data exchange are achieved by using Principal Component Analysis (PCA)-based data dimension reduction before transmission. The data are then sent to a third party, where they are processed using a semi-supervised K-means clustering algorithm. The resulting cluster labels are returned to the data owner, who performs label matching to localize potential failures.

The method's effectiveness is validated in terms of failure localization accuracy, achieving up to 98.5% on large-scale simulated datasets and 98% on small-scale experimental data.

1. Introduction

The growing complexity of modern communication networks, particularly optical networks, demands for automated, zero-touch management to ensure service continuity and performance. While hard failures (e.g., device outages or fiber cuts) result in immediate and severe service disruptions, soft failures, i.e., subtle degradations that may precede hard failures, offer a critical window for proactive intervention. However, localizing soft failures is notably challenging due to their heterogeneous and often ambiguous patterns [1].

In response, Machine learning (ML) techniques offer promising capabilities for enhancing network resilience through automatic and intelligent telemetry data analysis [2]. Yet, implementing such solutions demands considerable computational resources and expertise. Machine Learning as a Service (MLaaS) has emerged as a viable alternative, offering cost-effective and scalable access to ML tools [3]. Despite these advantages, MLaaS raises particular concerns about trustworthiness, particularly when sensitive operational data is outsourced to third-party providers for elaboration.

Thus, any soft failure management system exploiting MLaaS must reconcile the benefits of ML-driven insights with the need to safeguard critical data, which leads to several key system requirements: (i) accurate failure localization, (ii) real-time monitoring, (iii) data confidentiality assurance, and (iv) scalability across large and complex network environments. Previous studies have targeted some of these requirements [4] but did not consider them altogether.

The method proposed in this paper tries to do so by exploiting Principal Component Analysis (PCA) for confidentiality assurance and MLaaS-based clustering for soft failure localization. The peculiarities of the proposed method are as follows: the method contemporarily ensures confidentiality and scalability by means of PCA-based anonymization and dimensionality reduction; the method balances accuracy and data confidentiality assurance by applying a k-means clustering on selected PCA components at the third party and performing failure localization at the network provider; the method enables real-time failure localization by monitoring and analyzing network status data at every entry.

The conducted experimental testbed and simulation performance evaluation showed that the method is capable of localizing failures with a 98% accuracy for the experimental data and 98.5% for the simulated large-scale data. For the task of anomaly detection, an accuracy greater than 99% was achieved for all datasets.

2. State of the Art

Numerous studies have explored ML for failure management in optical networks [4], with a predominant focus on performance, particularly accuracy. Notably, the main goals of such systems, such as soft failure detection, identification, and localization, are reduced to the more basic task of anomaly detection [5] or, at most, hard failure forecasting [6]. More advanced capabilities like, identification, which distinguishes between failure types, and localization that pinpoints the exact network segment or device responsible [7], remain underexplored [4, 8].

As discussed in the introduction, collaborative management tools and MLaaS bring clear advantages but also raise significant concerns around trustworthiness. Ensuring any level of trust in such frameworks often entails compromises in model performance, particularly in latency or accuracy. Few studies tackle this issue. For instance, [9] uses homomorphic encryption for encrypted anomaly detection but lacks localization capabilities and real-time processing. [10] employs a federated learning approach for localization but without real-time support. Similarly, [11] and [12] propose confidentiality preserving techniques for offline detection or clustering, requiring full access to the dataset and excluding real-time applicability. [1] claims to offer a hybrid, scalable, and privacy-aware solution, yet its contributions would require a more thorough evaluation.

Real-time responsiveness, arguably the fundamental requirement of any monitoring tool, is among the least addressed aspects in the literature, even in non-confidentiality preserving scenarios. Among the surveyed works [9, 10, 11, 12, 13, 14, 15, 16, 17, 18], most approaches rely on processing cumulative historical data, i.e. batch processing, only [1] explicitly addresses real-time monitoring, highlighting a gap in existing research.

Another critical yet underexplored issue is scalability to realistic, large-scale network environments. Some studies rely on experimental dataset [9, 10, 11, 12], which offer greater realism, but they are typically constrained by small-scale testbeds and limited infrastructure. Other studies use simulated datasets [14, 15] which enable evaluation at scale but often fail to reflect the nuanced behaviors and complex dynamics of real-world optical networks.

In summary, although existing research reported in review papers [2, 4, 8] cover many advances, most studies address only a subset of the key requirements. A unified framework that combines scalable soft failure localization, privacy-preserving mechanisms, and real-time performance remains unexplored. Nonetheless, by carefully balancing trade-offs among detection accuracy, responsiveness, confidentiality, and scalability, developing a practical yet comprehensive ML-based failure management system is a feasible and necessary next step.

3. Proposed Methodology

Fig. 1 illustrates the methodology of the proposed approach. It involves two distinct phases: one occurring at the provider side and the other at the third-party side. As depicted in the upper

part of Fig. 1, an initialization phase is carried out by both parties. This happens when the proposed method is activated or whenever the network configuration changes, such as with new, unaddressed failures or reconfigurations. During the initialization phase, the provider gathers sample snapshots of the network parameters under various known failure states and normal operating condition, combining them into a reference (sample data) dataset. The reference dataset is structured as a matrix, where the columns represent network parameters (e.g., per-channel OSNR, BER, and amplifier input/output power levels) and the rows the time. Thus, the values contained in the cells represent the values at a time t_i of the network parameters (i.e., a network state snapshot at time t_i). For instance, if x failure states and one normal operating condition are considered, the provider collects $k = x + 1$ sample network snapshots into an initialization matrix K of dimension $k * j$, where j is the number of considered network parameters. Then, PCA is utilized to extract principal components (PCs) from the matrix K. Thus, the dataset is transformed into a lower-dimensional space (represented by a matrix with lower number of columns, e.g., 3 out of 7 columns) and sent to the third party. Although PCA obfuscates the data, each row still represents a specific observation of the network at a specific time.

In this approach, the third party, responsible for MLaaS, processes anonymized data received from the network operator every four seconds (corresponding to each new network observation). The third party then returns a code (that can only be inferred by the provider) representing a network state, i.e., indicating a specific failure at a particular network location (localization) or signifying a normal condition. By interpreting these codes, the provider can promptly identify failures, locate affected areas, and take necessary corrective actions.

Note that, if the dataset contains feature values respect to multiple types of failures occurring at the same location, the proposed method is also capable of performing identification.

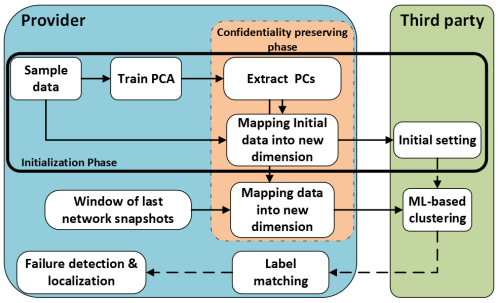


Figure 1. Proposed method workflow.

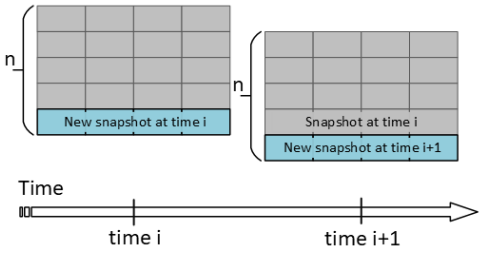


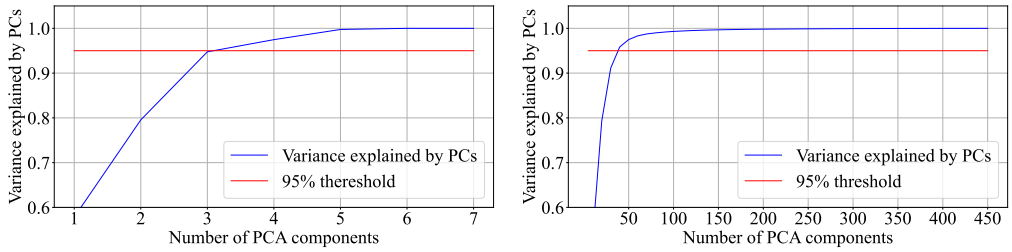
Figure 2. Sliding data window.

3.1. *Confidentiality Preserving Approach*

Data confidentiality is preserved with an anonymization method exploiting PCA. As noted by [19], PCA is a well-known method for dimension reduction to map a large dataset into a smaller one while retaining significant data patterns and trends through a so called transformation matrix. PCA replaces a set X original variables by a set Q latent variables. The set Q is called principal components (PCs) and is obtained by multiplying the original data matrix by the eigenvectors of its covariance matrix (transformation matrix). The dimension of Q is defined by the number of eigenvectors used, whether each PC retains a part of the variance of the original data. The quality of the Q set approximation can be measured by the sum of the variances associated with the retained PCs. The eigenvectors of the covariance matrix of X can be calculated using SVD. Applying SVD to X leads $X = UDV^\top$, from which the covariance matrix can be written as $X^\top X = VD^2V^\top$, where V are the eigenvectors and D^2 the eigenvalues associated with the eigenvectors [11].

Data confidentiality protection through PCA with noise addition is well known; however, the addition of noise reduces the utility for ML analysis. But, under specific conditions, even without

noise addition, PCA can significantly improve data confidentiality protection [20]. Because PCA is a lossy transformation that projects data onto a lower-dimensional space, reversing it without access to the transformation matrix is challenging. An attacker would need extensive knowledge of the original dataset to reliably approximate this matrix via, for example, a re-identification attack of known features. The accuracy of such an attack drops sharply when the amount of known features is reduced or the amount of dimensions/the complexity of the original data is increased. Hence, although PCA is not immune to reconstruction attacks, it demands increasingly higher levels of prior knowledge about the original data as the dimensionality and complexity of the original dataset increases. Thus, provided the original data has enough complexity and dimensionality or prior knowledge about the data is restricted, PCA can significantly increase data confidentiality protection, even without noise addition, allowing for its use in ML analysis.



(a) Experimental dataset. (b) Simulated dataset DB1.

Figure 3. Variance explained by PCs for two datasets.

The number of principal components (PCs) can be determined based on the variance they explain, ensuring it exceeds a required threshold or achieves the desired accuracy. For example, it will be shown that when PCs explain 95% of the variance, they result in an accuracy of over 90%. This relationship is further supported by the elbow rule, observed in the plot of variance explained by PCA as a function of the number of PCs, as depicted in Fig. 3.

3.2. Soft-Failure Localization

Soft-failure localization is achieved through a carefully designed workflow and the selection of a simple yet effective ML algorithm (i.e, k-means) capable of producing consistently labeled clusters from initialization.

Unsupervised clustering algorithms typically assign random names to clusters in each realization, even when perfectly differentiating network states (e.g., specific failure types and locations, or normal operation). This randomness can make it difficult for the provider to consistently recognize the failure type and its location.

To address this and increase the likelihood of correct failure localization, our method leverages K-means ability to perform clustering according to initial cluster samples (i.e., K). If initialized with these samples, K-means maintains the order defined in the K matrix. This allows the third party to perform the clustering, while the provider activates the initialization phase to ensure consistent labeling.

Therefore the third party utilizes the received matrix K to train a K-means clustering algorithm, by setting the number of clusters to k and using the k rows of the reduced dataset as cluster centroids. In this case, each cluster represents one of k considered network states, depicted with a specific code in the initialization phase. Each code represents a normal working condition or a failure type in a specific location. During the regular monitoring phase, once the anonymized data reaches the third party, clustering is performed and the code assigned to each snapshot of the network received is sent back to the provider.

3.3. Real-Time Processing

After finalizing the initialization in both parties, the provider conducts periodic data collection through a monitoring system (e.g., the one referenced in [21]). In this phase, data collection is based on the following mechanism: a new row (i.e., snapshot) is incrementally added to a fixed size monitoring matrix, while the oldest snapshot is removed, as depicted in Fig. 2. Each time the monitoring matrix is updated, the provider reduces it to the set of PCs and sends the obfuscated matrix to the third party to perform K-means. Each row of the matrix is labeled (i.e., assigned a code) with respect to one of the conditions known in the reference matrix K. By interpreting these codes (an action that can be done by means of information known by network provider only), the provider can promptly identify and locate failures and take necessary corrective actions. Considering only one new snapshot at a time exists in the window (the rest of the window are repeated, as depicted in Fig 2 ensures that, as soon as a new network snapshot is collected the third party can analyze it. In network monitoring, most of the time is spent on periodic data collection, primarily telemetry retrieval from network devices (near 4 seconds in our experiments), with Kafka transmission delays adding only a few tens of milliseconds. In contrast, data analysis and exchange with third parties, as performed per observation in our approach, are negligible (tens of milliseconds), enabling timely and effective real-time failure management.

4. Experiments

We evaluate the proposed method with two different types of dataset in order to assess its effectiveness in both real-time and scaled up situations. The first type is obtained from an experimental testbed, and the second type simulates a large scale optical network.

4.1. Experimental Test-bed Setup

We first evaluate the proposed approach on a dataset collected in a real testbed, shown in Fig 4. Three reconfigurable add-drop multiplexers (ROADMs) topology with two links of 80km each (L1 and L2) has been considered. Two DP-QPSK 100-Gb/s commercial muxponders are adopted to generate two WDM signals (Channel1 at 193.1 THz and Channel2 at 193.2 THz respectively). At the end node, the signals are collected by two coherent receivers.

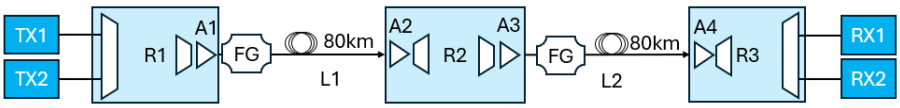


Figure 4. Considered testbed for data collection.

4.2. Experimental Data Generation

A Kafka-based monitoring tool [21] is adopted to collect the data from the optical devices, amplifiers (input and output power levels) and coherent receivers (OSNR and pre-fec BER). For each link, a failure generator module (FG) is exploited to realize perturbations on the transmission system. In particular, three types of failure are considered in the testbed: (i) attenuation only on channel1, (ii) attenuation only on channel2, (iii) attenuation affecting both channels. In this experiment, disjoint failures are generated separately within its respective link.

This approach results in the delineation of 6 failure states alongside a single state of normal operation (i.e., without any soft failure). Considering the monitored parameters, a dataset with 12 columns, including BER, OSNR, input power, and output power of the amplifiers and 4000 rows, corresponding to snapshots collected at 4-s intervals, is produced. The generated matrix pertains a monitoring period of four hours and forty-five minutes. Based on this baseline dataset, the matrix

K is produced with the 12 columns, and 7 rows representing six failure types and one normal operation corresponding to the labels from 0 to 6. Initialization is performed relied on matrix K. In regular monitoring phase, the window of size, 12 columns and 500 rows, slides through entire dataset, simulating practical network monitoring situation. The third party, receives the data every 4 seconds and replies with the corresponding labels to the transferred window of the provider.

4.3. Simulation Setup

The simulations are conducted on the NSFNet optical network topology [7] with a 4.8-THz optical wavelength band (C-band) and a 12.5-GHz grid. Demands are uniformly distributed across the network, with both symbol rate and modulation formats uniformly assigned. We assume symbol rates of 22, 34, 44, 56, and 68 GBaud and modulation formats of QPSK, 16QAM, and 32QAM. Routing and wavelength assignment use Dijkstra and first-fit algorithms. This setup creates a total of 1,316 bidirectional services, resulting in a utilization factor (UF) of 97.66%.

The transponders operate with polarization multiplexing and root-raised cosine shaping filters with $\alpha = 0.10$. The network employs route-and-select (R&S) ROADMs, each equipped with a per-channel power control loop using optical channel monitors (OCMs) and wavelength-selective switches (WSSs), maintaining a launch power per slot of 6 dBm. We assume 80-km spans with 0.2 dB/km attenuation, except for the last span, which ranges between 50 km and 120 km to reach the target total span length. Inline amplifier (ILA) gains and noise figures (NFs) are uniformly set to 16 dB and 5.5 dB, respectively.

4.4. Data Simulation

Soft failures are synthetically generated within a network digital twin (NDT) [22]. The NDT operates by capturing network snapshots where telemetry remains free of stochastic fluctuations. To incorporate time evolution, we simulate inline amplifier (ILA) gain degradations by gradually reducing amplifier gains. Each soft-failure scenario begins with a fault-free period lasting n snapshots, in which n is uniformly sampled between 5 and 9. The failure then gradually develops, with gain degradation increments drawn from a uniform distribution between 0.5 dB and 1 dB. The maximum degradation is also randomly selected from $\{1, 2, 3\}$ dB. For each amplifier fault, we collect a total of 20 snapshots, ensuring that only one ILA gain is affected per scenario. Thus, encompassing all NSFNet ILAs, 480 soft failures are computed in the NDT.

The final dataset consists of 480 files, each corresponding to an NSFNet ILA gain degradation. Each file contains 20 rows and 16,929 columns, where rows represent network snapshots and columns correspond to NDT telemetry data. Telemetry is collected from both amplifiers and transponders: for amplifiers, we include input and output power, while for transponders, we consider input and output power, GSNR, GOSNR, OSNR, and BER.

Concatenating all the files corresponding to the 480 simulated failure types, with 20 rows each, provides a first dataset (DB1) with 9600 rows and 16,929 columns. The second dataset (DB2) is produced in the same way, but adding 60 rows of normal condition per failure type yielding 38,400 rows and 16,929 columns. The third dataset (DB3) is produced in a similar fashion, 100 rows of normal condition, producing 57,600 rows and 16,929 columns. Based on different failures, the matrix K is produced with the 16,929 columns and 481 rows, representing 480 failure types and one normal operation corresponding to the labels from 0 to 480. Initialization is performed relying on matrix K. In the regular monitoring phase, window of 500 rows and 16,929 columns slides through entire dataset, simulating real network monitoring situation.

5. Results

The results for each dataset are evaluated from multiple perspectives, including accuracy and precision in soft failure localization, real-time monitoring, and scalability, as detailed below.

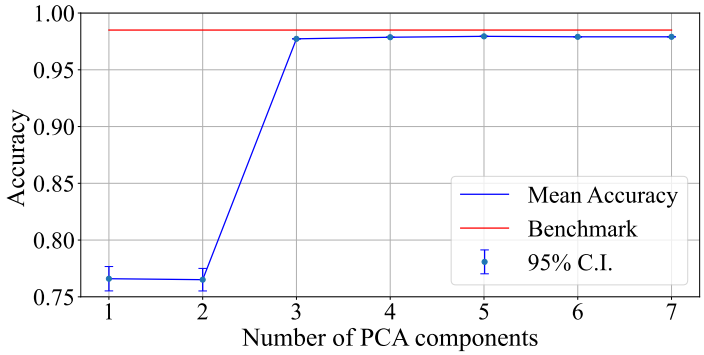


Figure 5. Accuracy as a function of PCs for experimental dataset.

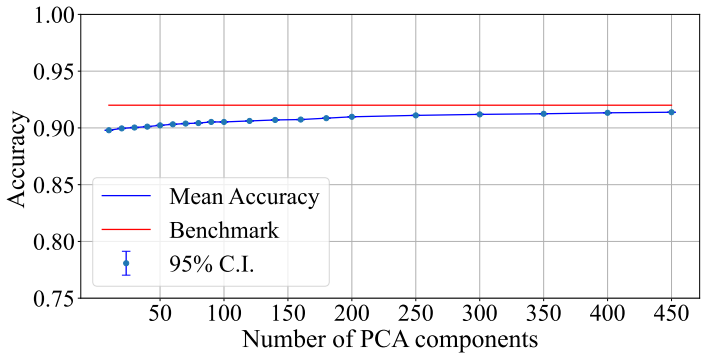


Figure 6. Accuracy as a function of PCs for simulated dataset DB1.

5.1. Accuracy

The accuracy of soft failure localization differs from that of anomaly detection. Although anomaly detection identifies any deviation from the normal operating state of the network, soft failure localization aims to pinpoint specific failure states. Soft-failure localization accuracy as a function of the number of components for the four implemented datasets is shown in figs. 5 to 8, with confidence interval of 95% and compared with benchmark line (i.e., without applying PCA). Increasing the number of components can improve accuracy; however, selecting the minimum number of PCs that yield acceptable accuracy helps to maintain confidentiality. Assuming 3 PCs out of 7 for the experimental dataset yields 98% accuracy in soft-failure localization while, for the simulated dataset, reducing from 16,929 features into 50 PCs results in 92% with DB1, 97% with DB2, and 98.5% with DB3 for localization accuracy and for the task of anomaly detection, an accuracy higher than 99% was achieved for all datasets.

As expected, the simulation results indicate that the accuracy levels change based on the density of failure states in the dataset, varying from DB1: 92% to DB3: 98.5%. This is because the transition from a normal operating state to a faulty state is not an instantaneous phenomenon and creates transient states that does not clearly belong to the normal or a specific failure state. The model is able to detect them as anomaly, but can't identify the failure type or localize their location. Accordingly DB1 contains a higher density of failure states compared to the experimental dataset, the corresponding accuracy level is lower, as depicted in Figs 5 and 6. However, when the density of the failure states decreases the accuracy level increases as shown Figs 7 and 8. Furthermore, the synthetic behavior of simulated dataset with respect to the

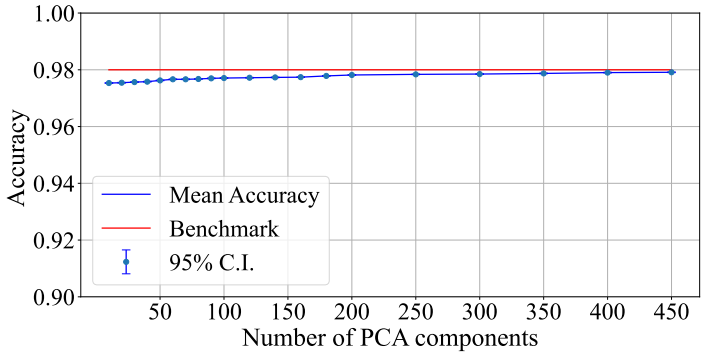


Figure 7. Accuracy as a function of PCs for simulated dataset DB2.

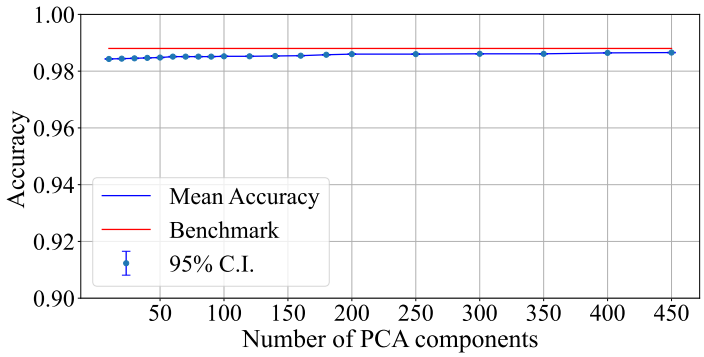


Figure 8. Accuracy as a function of PCs for simulated dataset DB3.

experimental one can be noticed through the low dependence on the number of PCA components.

5.2. Clustering Performance

We also evaluated standard metrics the performance of clustering algorithms, including *Homogeneity*, *Completeness*, *V-measure*, *Adjusted Rand Index* (ARI), and *Adjusted Mutual Information* (AMI) [23]. The average, 95% confidence interval of all metrics along with accuracy are computed by sliding a network snapshot window (of size 500 rows) across the entire dataset. The results presented in Figs 9 and 10 demonstrate that, by trading off a negligible performance, it is possible to achieve more confidentiality with significant dimension reduction. Specifically, using only 3 out of 7 components for the experimental data and 50 out of 16,929 components for the simulated data yields acceptable evaluation outcomes.

Based on this trade off, the high homogeneity and completeness scores (95%) indicate that the method not only groups similar failure types accurately but also maintains clear distinctions between different ones. Similarly, a V-measure greater than 95% reflects a well-balanced clustering performance, ensuring both precise classification and comprehensive failure detection. Furthermore, the high ARI and AMI scores confirm the strong alignment of clustering results with actual failure classification, demonstrating the method's ability to effectively capture and differentiate network state patterns.

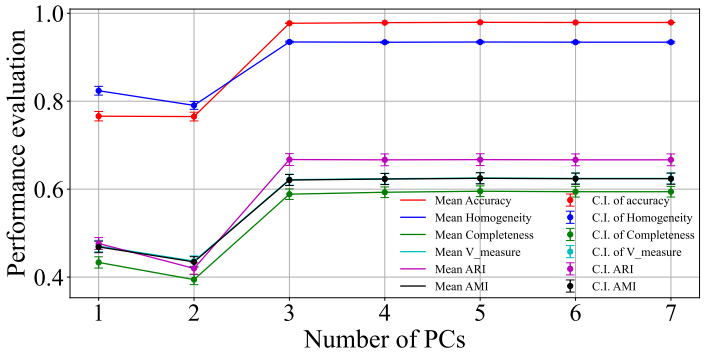


Figure 9. Performance evaluation with different metrics for experimental dataset.

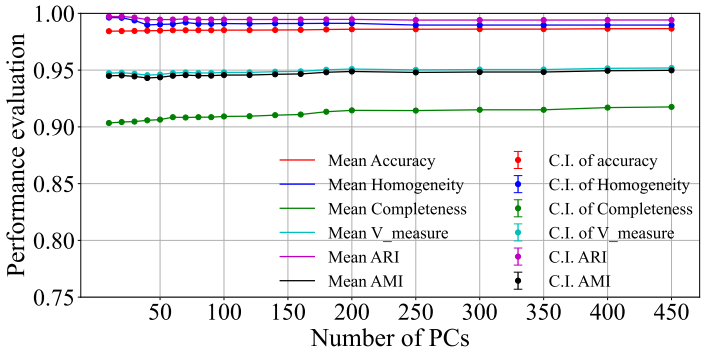


Figure 10. Performance evaluation with different metrics for simulated dataset DB3.

5.3. Precision

When using a sliding window of the last (e.g., 500 rows) network snapshots, each snapshot participates in multiple clustering processes equal to the number of rows in the sliding window. From the provider's perspective, it is undesirable to receive inconsistent failure classifications for the same network snapshot. Therefore, the model's results are specifically analyzed to ensure consistency and prevent any confusion on the provider's side.

To address this, we define a $precision$ metric, which quantifies the frequency of a correct and consistent classification assigned by the algorithm to a specific network snapshot. This metric tracks the clustering results for a snapshot throughout its presence in the sliding window. For example, in the scenario illustrated in Fig 2, where n=20, a network snapshot remains in the window and is labeled 20 times. Table 1 presents an excerpt of the precision results for five network snapshots. In the third row (i.e., t_{i+2}), for instance, a snapshot with a Ground Truth value of *0* is clustered as *0* in 19 out of 20 instances, achieving a precision rate of 95%. By iterating over the entire datasets, the average precision achieved is found to be >98% for both the experimental and simulated datasets.

5.4. Scalability and Real time Performance

Applying PCA, reduces the data transmission time by reducing the dimension. Considering a reduction from 16,929 to 50 columns strengthens not only the robustness of the data anonymization, but also enhances the data transmission. Assuming that the processing time includes the total time required for data scaling with PCA, and K-means clustering on 500 rows of raw data. In

Table 1. Tracing the *precision* of the method for a specific network snapshot

Network snapshot	1	2	3	...	20	Ground truth	Precision
t_i	0	0	0	...	0	*0*	100%
t_{i+1}	0	0	0	...	0	*0*	100%
t_{i+2}	4	0	0	...	0	*0*	95%
t_{i+3}	4	0	4	...	0	*0*	90%
t_{i+4}	4	4	4	...	4	*4*	100%

the same processor level, it takes an average of 0.062 seconds for experimental (small) data and 0.1437 seconds for DB3 simulated (large-scale) data (with 50 PCs), ensuring scalability and real-time capability, respect to network snapshot duration (i.e., 4 seconds).

Moreover, the results present advantages in terms of computation time when employing PCA. Indeed, for the largest dataset, i.e. DB3, the time needed to perform computations (PCA+K-means) is 0.1437 seconds (with 50 components) compared to 1.1870 seconds when PCA is not employed (i.e., only K-means). Hence, reducing the dataset dimension directly decreases data processing time.

6. Conclusion

This paper presents a novel method for collaborative soft failure localization in optical networks that combines real-time monitoring, high localization accuracy, and strong data confidentiality. By integrating PCA for dimensionality reduction and anonymization with a K-means clustering approach executed via MLaaS, the proposed solution effectively balances scalability and privacy.

Results on both experimental and simulated datasets proved that the method can achieve up to 98.5% accuracy in simulations and 98% on real-world data, even when reducing high-dimensional network observations to a minimal number of PCs. Future efforts will explore extending the method to support multi-failure scenarios, where multiple soft degradations may occur simultaneously. Additionally, we plan to investigate the integration of lightweight encryption techniques with PCA to further enhance confidentiality while maintaining the real-time performance of the MLaaS pipeline.

7. Acknowledgments

This work has been sponsored in part by the CHIPS-JU SMARTY project (GA 101140087) including top-up funding by the Italian Ministry of Enterprises and Made in Italy (MIMIT) and has been carried out also within the framework of the Department of Excellence in Robotics and Artificial Intelligence funded by the Italian Ministry of University, and Research (MUR). We would like to thank Arno van der Weijden from Technolution B.V. Netherlands for fruitful discussions about the security of PCA without noise addition for data confidentiality protection.

8. References

[1] Xiaoliang Chen et al. “Automating Optical Network Fault Management with Machine Learning”. In: *IEEE Commun. Mag.* 60.12 (2022), pp. 88–94. DOI: 10.1109/MCOM.003.2200110.

[2] Francesco Musumeci et al. “A Tutorial on Machine Learning for Failure Management in Optical Networks”. In: *J. Light. Technol.* 37.16 (2019), pp. 4125–4139.

[3] Mauro Ribeiro, Katarina Grolinger, and Miriam A.M. Capretz. “MLaaS: Machine Learning as a Service”. In: *2015 IEEE 14th International Conference on Machine Learning and Applications (ICMLA)*. 2015, pp. 896–902. DOI: 10.1109/ICMLA.2015.152.

[4] Rentao Gu, Zeyuan Yang, and Yuefeng Ji. “Machine learning for intelligent optical networks: A comprehensive survey”. In: *J. Netw. Comput. Appl.* 157 (2020), p. 102576. ISSN: 1084-8045. DOI: https://doi.org/10.1016/j.jnca.2020.102576. URL: https://www.sciencedirect.com/science/article/pii/S1084804520300503.

[5] Moises Felipe Silva et al. “Confidentiality-preserving machine learning algorithms for soft-failure detection in optical communication networks”. In: *J. Opt. Commun. Netw.* 15.8 (Aug. 2023), pp. C212–C222. DOI: 10.1364/JOCN.481690.

[6] Sadananda Behera, Tania Panayiotou, and Georgios Ellinas. “Machine learning framework for timely soft-failure detection and localization in elastic optical networks”. In: *J. Opt. Commun. Netw.* 15.10 (Oct. 2023), E74–E85. DOI: 10.1364/JOCN.490008.

[7] Kayol S. Mayer et al. “Machine-learning-based soft-failure localization with partial software-defined networking telemetry”. In: *J. Opt. Commun. Netw.* 13.10 (2021), E122–E131. DOI: 10.1364/JOCN.424654.

[8] Danshi Wang et al. “A review of machine learning-based failure management in optical networks”. In: *Sci. China Inf. Sci.* 65.11 (Oct. 2022). DOI: 10.1007/s11432-022-3557-9. URL: https://doi.org/10.1007/s11432-022-3557-9.

[9] Xiaoqin Pan, Shaofei Tang, and Zuqing Zhu. “Privacy-Preserving Multilayer In-Band Network Telemetry and Data Analytics”. In: *2020 IEEE/CIC International Conference on Communications in China (ICCC)*. 2020, pp. 142–147. DOI: 10.1109/ICCC49849.2020.9238883.

[10] Memedhe Ibrahimi et al. “Vertical Federated Learning for Failure Localization in Partially Disaggregated Optical Networks”. In: *2024 IEEE 25th International Conference on High Performance Switching and Routing (HPSR)*. 2024, pp. 1–6. DOI: 10.1109/HPSR62440.2024.10635921.

[11] R. F. Sales et al. “Disaggregated Confidentiality-Preserving Scheme for Fault Detection in Optical Networks”. In: *2024 Optical Fiber Communications Conference and Exhibition (OFC)*. 2024, pp. 1–3.

[12] Azarm Yeganehfallah et al. “Effectiveness of Confidentiality-Preserving Clustering Algorithms for Soft Failure Detection in Optical Networks”. In: *2024 IEEE 25th International Conference on High Performance Switching and Routing (HPSR)*. Pisa, Italy, 2024, pp. 87–92. DOI: 10.1109/HPSR62440.2024.10636007.

[13] M. F. Silva et al. “Confidential Detection of Multiple Failures in Optical Networks: an Experimental Evaluation”. In: *Optical Fiber Communication Conference (OFC) 2023*. Optica Publishing Group, 2023, Th2A.16. DOI: 10.1364/OFC.2023.Th2A.16. URL: https://opg.optica.org/abstract.cfm?URI=OFC-2023-Th2A.16.

[14] Nazila Hashemi et al. “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. DOI: 10.1109/ECOC52684.2021.9605846.

[15] Kayol S. Mayer et al. “Soft Failure Localization Using Machine Learning with SDN-based Network-wide Telemetry”. In: *2020 European Conference on Optical Communications (ECOC)*. 2020, pp. 1–4. DOI: 10.1109/ECOC48923.2020.9333313.

[16] T. Panayiotou, S. P. Chatzis, and G. Ellinas. “Leveraging statistical machine learning to address failure localization in optical networks”. In: *J. Opt. Commun. Netw.* 10.3 (2018), pp. 162–173. DOI: 10.1364/JOCN.10.000162.

[17] Shahin Shahkarami et al. “Machine-Learning-Based Soft-Failure Detection and Identification in Optical Networks”. In: *2018 Optical Fiber Communications Conference and Exposition (OFC)*. 2018, pp. 1–3.

[18] Zhilong Wang et al. “Failure prediction using machine learning and time series in optical network”. In: *Opt. Express* 25.16 (Aug. 2017), pp. 18553–18565. DOI: 10.1364/OE.25.018553. URL: https://opg.optica.org/oe/abstract.cfm?URI=oe-25-16-18553.

[19] I.T. Jolliffe. *Principal Component Analysis*. Springer Verlag, 1986.

[20] Hiromi Yamashiro et al. “A study of the privacy perspective on principal component analysis via a realistic attack model”. In: *2022 18th International Conference on Computational Intelligence and Security (CIS)*. 2022, pp. 376–380. DOI: 10.1109/CIS58238.2022.00085.

[21] Andrea Sgambelluri et al. “Reliable and scalable Kafka-based framework for optical network telemetry”. In: *J. Opt. Commun. Netw.* 13 (Oct. 2021), E42–E52. DOI: 10.1364/JOCN.424639.

[22] Kayol S. Mayer et al. “Network-Wide QoT Estimation With Optimized Gradient Transfer Between Wavelengths”. In: *J. Light. Technol.* (2025), pp. 1–9. DOI: 10.1109/JLT.2025.3538951.

[23] Fabian Pedregosa et al. “Scikit-learn: Machine learning in Python”. In: *J. machine learning research* 12 (Oct. 2011), pp. 2825–2830.