Data-Constrained Semi-Supervised Approaches for Optical Network Fault Detection

Adryele Oliveira, Giovana Nascimento, Andrei Ribeiro, Fabrício Lobato, Moisés Silva, and João C.W.A. Costa.

Abstract— **With the emergence of 6G and IoT systems, it becomes crucial to correctly detect faults in optical networks to guarantee access to these vital systems. Although machine learning approaches show promise, most demand large datasets, often scarce in practice, posing a significant challenge for model deployment. In this work, we evaluate three semi-supervised learning approaches specifically selected for their complementary strengths in data-scarce scenarios: Principal Component Analysis (PCA) for its noise-resistant dimensionality reduction, One-Class SVM (OCSVM) for its robust boundary learning with limited normal samples, and Hierarchical Clustering (HC) for its adaptability to network operation patterns. All models are trained exclusively on normal operation data and progressive data reductions to assess their performance in resource-constrained scenarios. Experimental results using optical testbed telemetry data show that PCA, OCSVM, and HC achieve accuracies of 93%, 91.91%, and 74.31%, respectively, when trained with only 5% (544 samples).**

Keywords— **Optical Networks, Machine Learning, Failure Detection, Reduced Training Data.**

I. Introduction

As several data-hungry online applications (e.g., 6G systems, generative AI-driven applications) are strongly dependent on the ultra-high transmission capacity of optical networks, ensuring their reliability becomes critical to fulfilling practical requirements [1]. This shift elevates reliability from a desirable feature to a fundamental operational necessity, as modern digital infrastructure must deliver uninterrupted services to meet the increasing performance demands. The importance of fault resilience becomes evident when considering the cascading effects of network failures (e.g., fiber cut, dirty connector, equipment aging), which can lead to significant service disruptions and widespread economic impacts. These systemic risks highlight the need for advanced detection mechanisms to identify failures [2].

Traditional simplified threshold-based methods [3] struggle to adapt to the dynamic and complex nature of modern optical networks, often resulting in false alarms or delayed responses. As a consequence, this motivates the adoption of Machine Learning (ML) approaches, with Supervised Learning (SL) methods demonstrating robust performance when sufficient labeled failure data is available, as shown in [4]. In contrast, Semi-supervised learning (SSL) addresses label scarcity by training mostly on unlabeled normal data, augmented with a small set of labeled samples. Thus, it is ideal for fault detection when anomalies are rare but normal operation data is abundant.

However, beyond the inviability of failure data, a critical challenge in optical network management is the limited availability of training data during the initial deployment of fault detection systems. Operational networks frequently encounter scarce historical data, creating fundamental constraints for ML applications. This data scarcity stems from the rarity of labeled failure events (costly to obtain) and the rapid obsolescence of older datasets due to changing network conditions. Consequently, reducing the required training data volume becomes particularly relevant for optical networks. These limitations have motivated the adoption of SSL-based approaches, such as Principal Component Analysis (PCA), One-Class Support Vector Machines (OCSVM), and Hierarchical Clustering (HC), that can perform effective fault detection in resource-constrained environments. They are advantageous not only for not requiring labeled failure data, but also for establishing robust operational baselines, even when trained on low-data regimes [5].

In that regard, in this work, we compare the failure detection performance of three SSL techniques, named as PCA, OCSVM, and HC, under progressively reduced training data. In addition to implementing fault detection in resource-limited scenarios, the analyses contribute to the model deployment of each model in these scenarios.

II. Theorethical background

A. Principal Component Analysis

PCA is a widely used technique in ML and statistics for dimensionality reduction and data visualization. It can reduce the data dimensionality by transforming it into a new coordinate system, where the variables are uncorrelated, orthogonal, and ordered by the amount of variance they capture [6]. This 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. The centered data matrix is denoted as X', from which the covariance matrix is computed. Each element of this matrix represents the covariance between pairs of variables, forming a square symmetric matrix. The eigenvector $\mathbf{v}$ of a covariance matrix $\mathbf{A}$ in PCA is found by solving the characteristic equation:

Adryele Oliveira, Giovana Nascimento, Andrei Ribeiro, Fabrício Lobato, and João CWA Costa, Insitute of Technology, Federal University of Pará, Belém-Pará, Brazil, e-mail: [adryele.oliveira, giovana.nascimento.silva, andrei.ribeiro]@itec.ufpa.br, [frl, jweyl]@ufpa.br; Moisés F. Silva, Los Alamos National Laboratory, Los Alamos-NM, USA, e-mail: mfelipe@lanl.gov

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

where λ is the eigenvalue corresponding to $\mathbf{v}$. Eigenvalues λ are obtained 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 $\mathbf{v_1}$, $\mathbf{v_2}$, ... $\mathbf{v_p}$ and eigenvalues $\lambda_1, \lambda_2, ..., \lambda_p$ of $\mathbf{A}$. These eigenvectors form the new basis vectors of the transformed coordinate system.

Finally, PCA selects a subset of the eigenvectors, known as principal components (PCs), corresponding to the k largest eigenvalues. These PCs represent the most important directions of variation in the data. The optimal number of PCs k is typically selected by analyzing the explained variance ratio, which quantifies the amount of information that each component retains from the original data [7].

B. One-Class Support Vector Machine

Support Vector Machine (SVM) is an ML technique used for classification and regression problems, and is widely applied in fault detection due to its strong generalization capability in handling nonlinear data effectively [8]. One-Class SVM (OCSVM) is a variation of SVM designed to train using only positive information (normal data). As a kernel-based method, given training data $\mathbf{X} = \{\mathbf{x_1}, \mathbf{x_2}, \ldots, \mathbf{x_l}\}$, a compact subset of $\mathbf{R^n}$, the algorithm maps them into a higher-dimensional feature space H ($\mathbf{\Phi}(\mathbf{x}) : \mathbf{X} \to \mathbf{H}$) via a kernel function f. In this H dimension, a maximum margin hyperplane ($\mathbf{w} \cdot \mathbf{\Phi}(\mathbf{x_i}) = \rho$) defined by the support vectors separates the data from the origin. So, the training data is the first class, and the origin is the only member of the second class. To achieve this separation, a quadratic programming problem must be solved [8]:

$$f(w) = \min \frac{\|w\|^2}{2} - \rho + \frac{1}{\nu n} \sum_{i=1}^{n} \xi_i, \tag{3}$$

subject to $\omega \cdot \mathbf{\Phi}(\mathbf{x_i}) \geqslant \rho - \xi_\mathbf{i}$, $\xi_\mathbf{t} \geqslant \mathbf{0}$. Normal data are contained within this high-density region inside the hyperplane w and equal to +1 by the function f, while the anomalies are in the sparse region and equal to -1.

C. Hierarchical Clustering

Clustering is the process of merging data into groups based on similarity. Each cluster contains samples that are highly similar to each other (high intra-cluster similarity) and significantly different from samples in other clusters (low inter-cluster similarity). Among the various clustering approaches, Hierarchical Clustering (HC) is a technique that builds a multilevel hierarchy of clusters. HC has two main approaches which define how this hierarchy will be formed: bottom-up (agglomerative clustering) and top-down (divisive clustering) [9].

In agglomerative clustering, the algorithm starts by merging single-point clusters. At each step, it then merges multipoint clusters with either single-point or multipoint clusters, continuing until a single group containing all the data is formed, creating a binary tree structure called a dendrogram. The user selects the number of clusters and specifies where the binary tree will be divided. The dissimilarity between clusters is determined by three main linkage criteria: average link (measures the average distance between all pairs), single link (defined as the distance between the two closest members of each group), and complete link (defined as the distance between the two most distant pairs). In practice, the most commonly used method in the literature is average link clustering [9], calculated as follows:

$$davg(G,H) = \frac{1}{n_G n_H} \sum_{i \in G} \sum_{i \in H} d_{i,i'}, \tag{4}$$

where $\mathbf{n_G}$ and $\mathbf{n_H}$ are the number of elements in groups G and H. After inducing a clustering of a given size, the centroid of each cluster is calculated based on the Kernel Density Estimation (KDE), given by:

$$\hat{p} = \frac{1}{n} \sum_{i=1}^{n} K_h\left(x - x_i\right), \tag{5}$$

where bandwidth h determines how smooth the density estimate will be, and kernel $\mathbf{k_h}$ defines the shape of the smoothing function.

III. Failure Detection approach

Fig. 1 presents the complete workflow of the proposed approach, encompassing both training and testing phases. Since the method operates in an SSL manner, only data from normal network operations are used during the training phase. To assess how the availability of data impacts the performance of the models, the training is performed by varying data reduction percentages - 100%, 80%, 60%, 40%, 20%, 10%, 5%, 1%, 0.5%, and 0.1%. As a result, it is possible to analyze the changes in classification accuracy under limited training conditions.

Firstly, by learning hidden patterns from this data, the models picture the inherent characteristics of normal conditions. In the OCSVM and HC models, Failure Indicators (FIs) are generated (as reconstruction errors in PCA, distances to centroids in HC, and distances to support vectors in OCSVM), and anomaly thresholds are established based on the FI values obtained from the training set, ensuring that only significant deviations from normal patterns trigger fault detection. In contrast, the PCA model is trained to reconstruct this data in their output back to the original feature space, m, after reducing to the k PCs. The aim is to minimize the MSEs, which measure how accurately the data was recreated. At the end of training, MSEs are typically small, and any remaining variances are considered acceptable.

During the testing phase, consisting of both normal and anomalous data, the FIs are calculated, which quantify how much a test sample deviates from the learned representation of normal behavior. A sample is classified as a failure if at

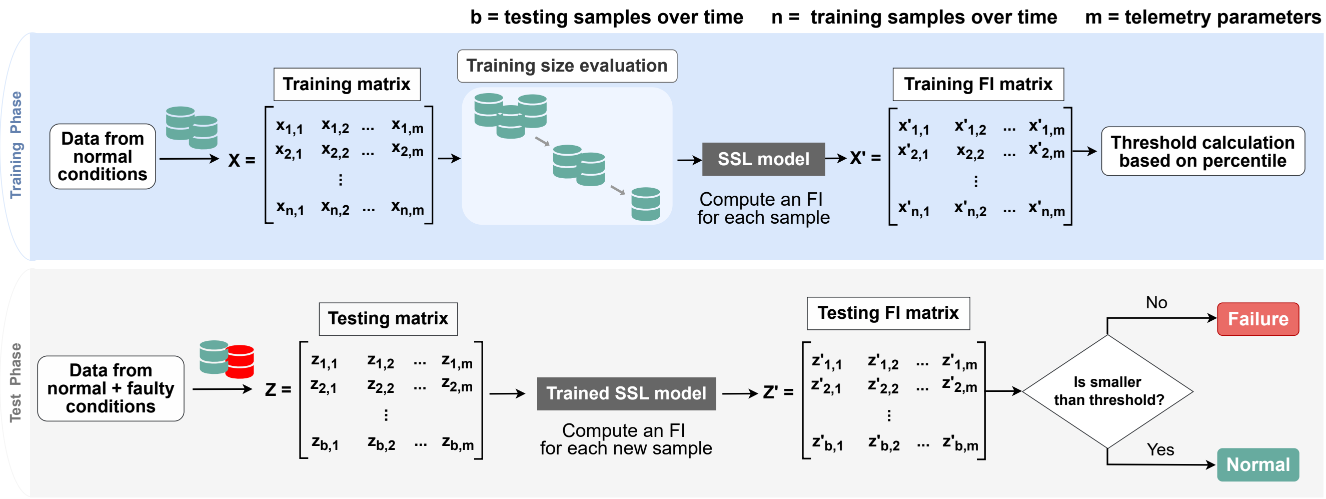

b = testing samples over time
n = training samples over time
m = telemetry parameters
Training Phase
Data from normal conditions
X =
Training matrix
Training size evaluation
SSL model
Compute an FI for each sample
X' =
Training FI matrix
Threshold calculation based on percentile
Test Phase
Data from normal + faulty conditions
Z =
Testing matrix
Trained SSL model
Compute an FI for each new sample
Z' =
Testing FI matrix
Is smaller than threshold?
No
Failure
Yes
Normal

Fig. 1: Overview of the proposed approach.

least one of its FIs exceeds the threshold defined in the training phase; otherwise, it is considered normal. In general, OCSVM and HC FIs are computed as the Euclidean distance between each test sample and a set of reference points derived during training. These reference points differ depending on the model: in the OCSVM, they correspond to the support vectors that define the decision boundary of normal data; in the HC model, they represent the density peaks of each cluster. On the other hand, PCA computes the reconstruction errors and compares them to the given threshold value for actual failure detection, created based on a chosen percentile value that limits the MSEs of the sorted vectors.

IV. Results

A. Experimental Setup and Parameter Definition

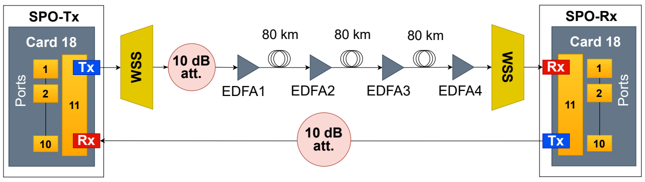


Fig. 2: Optical network testbed.

To evaluate the proposed approach, a publicly available dataset from GitHub was employed, which is based on the telemetry described in [10]. The monitoring system, as shown in Fig.2, includes two Ericsson SPO 1400 devices, one Wavelength Selective Switch (WSS), and four EDFA amplifiers. All the EDFA amplifiers are controlled via SPO devices (ampli1 and ampli2 are controlled by SPO-Tx, while SPO-Rx controls ampli3 and ampli4) and present a configurable gain in the range 15-25 dB, with output mute power of 0.4 dBm. All the amplifiers are configured in constant gain mode, with a gain value that allows each span to be entered with 0 dBm of optical power. Each SPO is equipped with a 100 Gb/s Optical Transport Network (OTN) muxponder (installed at slot 18) with a DWDM optical line (port 11) and 10 tributary ports. The output of the first SPO (SPO-Tx) has been attached to a WSS, which is then attached to a multi-span link over a 10 dB attenuator. The optical link between the SPO-Tx and SPO-Rx consists of three spans, each spanning 80 km. The data is collected every 3.5 seconds over 10 hours.

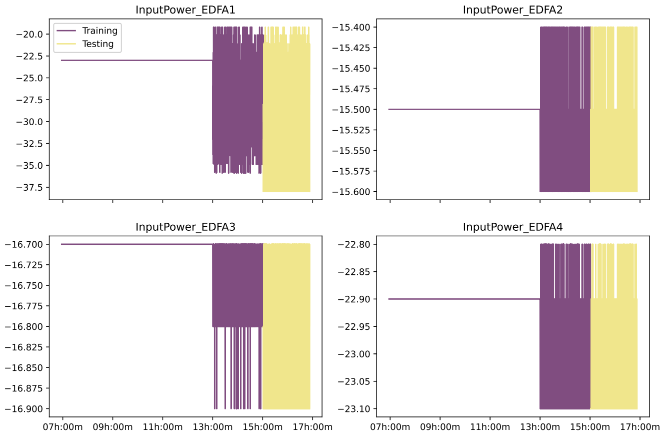


Fig. 3: Features distributions.

In the first 8 hours, two normal operation conditions were simulated, as shown in Fig.3: a stationary normal behavior during the first 6 hours, and a noisy normal behavior in the remaining 2 hours by randomly changing the attenuation in the range from 0 to 18 dB. In the remaining 2 hours, the same behavior as during the last 8 hours is simulated, but with a 25 dB attenuation added every 40 seconds, which puts the network in a failure condition for 10 seconds. Due to missing values in the raw dataset, an interpolation technique was applied, resulting in a final dataset of 13,948 samples. The dataset was partitioned into training and testing sets, with the first 80% (10,884 samples) of the data used for training and the remaining 20% (3,064 samples) reserved for testing.

The proposed approach initially trains each ML technique using 100% of the available training data to establish a baseline performance. Then, to investigate the impact of reduced training data on model performance, progressively decreased the number of training samples by randomizing the dataset and selecting subsets corresponding to 80% (8,707 samples), 60% (6,530 samples), 40% (4,353 samples), 20% (2,176 samples), 10% (1,088 samples), 5% (544 samples), 1% (108 samples), 0.5% (54 samples), and 0.1% (10 samples) of the original training set. This systematic reduction allowed to observe the gradual changes in classification performance for a constant number of test data (3,064 samples). The performances of the models are evaluated using accuracy, Type I and Type II error indications.

Moreover, the randomization process ensured that the subsets included a balanced amount of samples from both stationary and noisy normal operation conditions to maintain the balance of the training process. Each technique is configured with specific parameters to optimize the anomaly detection performance while maintaining computational efficiency.

Several trials were performed based on the best configuration of each model. For the PCA model, the input dimensionality was reduced to a single principal component (k=1), capturing the most significant variance pattern in the EDFA input power measurements. Meanwhile, the OCSVM implementation used an RBF kernel, where the parameter ν was set to 0.002 to control the tightness of the decision boundary around normal operation data, and the parameter γ was fixed at 1.0 to determine the influence radius of each support vector. HC adopted an agglomerative approach for the clustering component with average linkage and 8 clusters, determined through silhouette analysis. KDE with Gaussian kernels (h=0.01) identified density peaks within each cluster. A detection threshold was established at the 99th percentile of the error distribution from the training set, classifying any sample exceeding this boundary as a potential failure.

B. Comparison Results

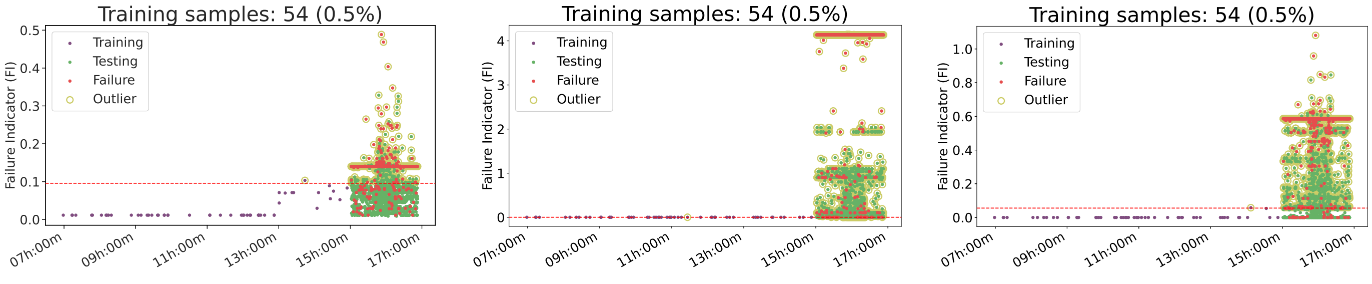
In this subsection, the impact of training data availability on failure detection performance is analyzed. According to the results in table I, it can be seen that OCSVM maintains a low Type II error as the training data is reduced; however, its Type I error increases significantly, showing the highest overall Type I error compared to the other models. Furthermore, it is evident that OCSVM performs better when a larger amount of training data is available. Regarding HC, it presents the weakest overall performance across all training data percentages and generally the highest Type II error rates. Despite outperforming OCSVM in 1% and 0.5% training data scenarios, indicating that even though it can be competitive, its high sensitivity to data volume still limits its suitability for practical deployment where both error types are critical.

Training data (%)	PCA		HC		OCSVM	
	Type I (%)	Type II(%)	Type I (%)	Type II (%)	Type I (%)	Type II (%)
100	3.40	3.16	6.7559	2.611	**1.1423**	**3.3616**
80	1.86	3.33	6.7559	2.611	**1.2402**	**3.2963**
60	**0.70**	**3.56**	6.9191	2.5783	3.2311	3.4269
40	**0.70**	**3.53**	3.0352	22.1606	2.3499	3.3616
20	**0.70**	**3.53**	2.154	21.8995	3.1658	3.3943
10	3.59	3.46	4.047	21.8668	**1.8603**	**3.1332**
5	**3.69**	**3.30**	3.8512	21.8342	5.3198	2.7742
1	**6.04**	**3.17**	2.6762	22.1279	35.7376	1.436
0.5	**5.02**	**4.98**	37.5653	1.3708	58.2898	0.718
0.1	**64.09**	**0.62**	73.3029	0.1305	73.5313	0.1305

TABLE I: Percentile of erros for PCA, HC, and OCSVM. Bold values indicate the best performance at each training data size.

The high Type I errors (>70%) observed for all methods at 0.1% training data (10 samples) demonstrate a clear data scarcity threshold. At this minimal data volume, models classify most samples as anomalies due to insufficient normal behavior characterization. This range was intentionally included to test the absolute lower limits of PCA, demonstrating that even with just 10 samples, it maintains reasonable Type II error rates (0.62%) while Type I errors escalated. This behavior suggests that PCA is highly sensitive to outliers in the test phase. However, in practical implementations, one must consider that high false alarm rates could increase the operational demands regarding network inspection. The sharp performance improvement at 0.5% data (Type I errors dropping to 5.02% for PCA) suggests this is a more realistic trade-off for operational deployments. At the same time, the 0.1% case serves primarily to establish experimental performance boundaries.

Accordingly, Fig. 4 illustrates the performance of each model under data scarcity conditions, plotting the failure indicator over time. The training data set (purple) consist solely of normal data, while the testing data set includes data from both normal (green) and failure conditions (red), with detected failures (outliers) marked by yellow circles. Overall, the results show PCA maintains superior separation, with most normal samples below the established training threshold and failures above it, indicating lower Type I and II error rates. In contrast, OCSVM and HC exhibit poor discrimination, with excessive failure samples below the threshold, reflecting higher Type I errors despite equivalent training data. This demonstrates their inability to properly distinguish anomalies from normal



(a) PCA

(b) OCSVM

(c) HC

Fig. 4: Failure detection performance of the proposed approaches.

operation points to fundamental limitations in their one-class classification capabilities under these experimental conditions.

C. Results Analysis

By analyzing Table I, Fig. 4, and Fig. 5, it is possible to extract some insights about the limitations and capabilities of the evaluated techniques. For instance, if the analysis were restricted to larger dataset scenarios, such as 100%-80% training data range, OCSVM would appear to be the best-performing method. However, when evaluating performance across the entire range of training data percentages, PCA demonstrates greater overall stability and robustness, not only maintaining a stable accuracy of approximately 90% throughout the range down to 0.5%, but also showing the lowest percentile of errors in 0.1% training data, demonstrating its resilience to limited data. In contrast, HC shows gradual improvement and reaches competitive accuracy levels only with larger datasets, achieving its last high accuracy at 60% training data, its outliers shown in Fig. 5, and afterward the accuracy stabilizes around 70%.

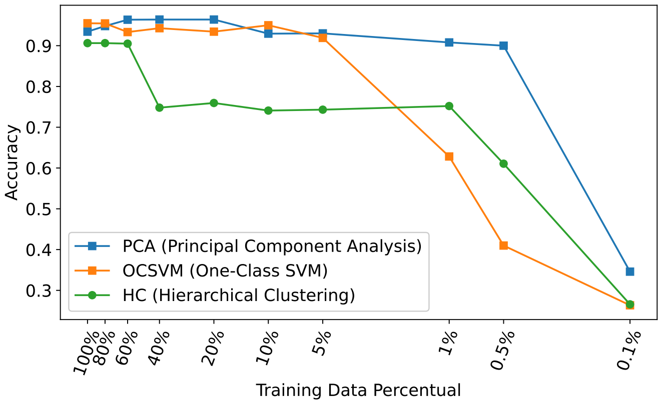


Fig. 5: Fault detection accuracy of the compared models per training data percentage.

From the above comparative test results, PCA demonstrates superior stability with limited training data because its dimensionality reduction approach fundamentally captures the most significant variance patterns in the feature space, which tend to remain consistent even with small sample sizes. PCA effectively filters out high-frequency noise by projecting data onto orthogonal principal components while preserving the low-dimensional manifold that characterizes normal network operation. This noise resilience stems from the method's reliance on global covariance structures rather than local data density, a property that makes it less sensitive to exact sample counts compared to distance-based methods like HC or boundary-sensitive approaches like OCSVM. Furthermore, the reconstruction error metric naturally normalizes for data volume, as it measures deviations relative to the dominant variance directions learned during training. This explains why PCA maintains about 90% accuracy down to 0.5% training data while other methods degrade sharply, making it particularly suitable for applications where historical fault data is scarce but the fundamental physics of optical signal propagation (and thus feature correlations) remain stable.

These results suggest that PCA is the most robust technique for scenarios where data collection is constrained, ensuring consistent performance even with a small number of training samples. On the other hand, OCSVM proves to be effective when larger datasets are available, and HC requires sufficient data to achieve reliable failure detection.

V. Conclusion

In this work, three SSL-based approaches (PCA, OCSVM and HC) are evaluated in reduced data scenarios. The comparative study demonstrates that PCA emerged as the most robust technique for optical network fault detection in conditions of limited training data, maintaining 90% accuracy with just 0.5% training data and minimal Type II errors, which are critical for failure prevention. On the other hand, OCSVM achieves peak performance (97.8% accuracy) with full datasets while it degrades significantly below 10% training samples. HC proves to be the least effective in low-data scenarios, requiring more than 60% data to reduce Type II errors below 20%. The results demonstrate that PCA is suitable for resource-constrained deployments due to its noise-resistant dimensionality reduction, while OCSVM remains a viable option when in scenarios with abundant training data.

Acknowledgments

This work was supported by the National Council for Scientific and Technological Development (CNPq) and by Amazon Sustainability Institute with Science and Innovation (ISACI). Thanks to Andrea Sgambelluri and Luca Valcarenghi from Scuola Superiore Sant'Anna for making the dataset available.

References

[1] W. Wang, M. Tornatore, and B. Mukherjee, "Machine Learning for Network Automation: Overview, Architecture, and Applications", *IEEE Communications Surveys & Tutorials*, vol. 24, no. 2, pp. 1208–1247, 2022.

[2] A. Ferrari, P. Barletta, and V. Curri, "6G and Optical Networks: Challenges and Opportunities", *IEEE Access*, vol. 10, pp. 121456–121478, 2022.

[3] D. Masters, C. S. Hood, and N. S. V. Rao, "Practical Machine Learning for Network Automation: Algorithms and Use Cases", *IEEE Network*, vol. 34, no. 3, pp. 20–26, 2020.

[4] J. Yoon, R. R. Fontugne, and K. Cho, "Few-Shot Learning for Network Anomaly Detection", *IEEE Journal on Selected Areas in Communications*, vol. 40, no. 5, pp. 1600–1614, 2022.

[5] G. Rizzelli, M. Fiorani, and L. Wosinska, "Unsupervised Clustering for Fault Detection in Optical Networks", *IEEE Transactions on Network and Service Management*, vol. 19, no. 3, pp. 2456–2470, 2022.

[6] A. A. Mohammed et al., "Principal Component Analysis for Anomaly Detection in Optical Networks: Theory and Experimental Validation", *Optics Express*, vol. 28, no. 6, pp. 7892-7910, 2020.

[7] Y. Zhao et al., "Real-Time Anomaly Detection Using PCA for Optical Network Monitoring", *J. Lightwave Technol.*, vol. 39, no. 15, pp. 5122-5135, 2021.

[8] B. Schölkopf, J. Platt, J. Shawe-Taylor, A. J. Smola, and R. C. Williamson, "Estimating the Support of a High-Dimensional Distribution," *Neural Computation*, vol. 13, no. 7, pp. 1443–1471, 2001.

[9] E. Arias-Castro and E. Coda, "An Axiomatic Definition of Hierarchical Clustering," *arXiv*, 2024. Available: https://arxiv.org/abs/2407.03574.

[10] InRete Lab, Optical Failure Dataset. Scuola Superiore Sant'Anna, 2021. [Online]. Available: https://github.com/Network-And-Services/optical-failure-dataset