Integrating Feature Extraction with Semi-Supervised Learning for Failure Detection in Optical Networks

Vitor Dias, Marcílio Santos, Andrei Ribeiro, Fabrício R. Lobato, Moisés F. Silva and João C. W. A. Costa.

***Abstract*— Since optical networks support data-intensive applications, ensuring transmission reliability is crucial. Conventional failure detection often relies on simplified thresholds or supervised learning, requiring large failure datasets. Semi-supervised methods offer viable alternatives by handling limited or imbalanced data. This work proposes a failure detection method integrating an autoregressive (AR) model and a one-class support vector machine (OCSVM). The AR model extracts relevant features, enhancing OCSVM performance. The approach was evaluated using an optical testbed dataset. Results show that integrating AR with OCSVM improved detection accuracy from 66.86% to 84.17% compared to the traditional OCSVM without AR.**

***Keywords*— Optical Networks, Semi-supervised Learning, One-Class Support Vector Machine, Autoregressive Model, Failure Detection.**

I. Introduction

A wide range of services and applications (e.g., 6G systems and generative AI-driven applications) increasingly depend on fast and reliable data transmission from optical networks. Therefore, ensuring the quality of transmission of these systems becomes crucial, as the optical networks are the only technology capable of meeting these data requirements. Moreover, optical networks are susceptible to faults that lead to packet loss or service degradation, making accurate fault detection methods vital [1].

Typically, conventional fault detection methods rely on simplified thresholds, which exhibit limitations in practical optical networks with many parameters [2]. Hence, machine learning (ML)-based approaches enable handling complex tasks and support effective and automated fault detection [3]. Most ML-based approaches focus on supervised learning (SL) algorithms. However, SL algorithms require a large amount of data from normal and fault conditions to be adequately trained. This requirement hinders their deployment in practical scenarios, as fault data is often scarce and hard to collect [4].

In contrast, semi-supervised learning (SSL)-based approaches are a promising alternative, as they require only normal data for training. In that case, the model can effectively recognize data from failure conditions once fed with it [5]. Recent studies demonstrate the applicability of the one-class support vector machine (OCSVM) algorithm in optical networks. Patri et al. [6] compare the effectiveness of the OCSVM and artificial neural network (ANN) algorithms for fault detection in optical networks under the optical spectrum-as-a-service (OSaaS) model. Similarly, Abdeli et al. [12] uses OCSVM applied to fault detection in an optical network to compare the effectiveness of an autoencoder-based approach with Gated Recurrent Units (GRUs)

Therefore, in this work, we propose an SSL-based approach that combines a One-Class Support Vector Machine (OCSVM) algorithm with an autoregressive (AR) model to detect faults in optical networks. The AR model extracts essential information from the data, enhancing the learning of OCSVM using normal operating conditions. Proper mapping of these conditions may enable the proposed approach to detect various types of failures accurately. To date, no studies have been conducted on the application of AR to optical networks.

The rest of the paper is structured as follows. Section II presents the theoretical fundamentals of the OCSVM and the AR model. In Section III, the results of the proposed approach are presented. Finally, in Section IV, the conclusions are presented.

II. AR-Assisted OCSVM-based Approach for Failure Detection

A. One-Class Support Vector Machine

Support Vector Machine (SVM) is a supervised ML technique for classification and regression problems. It is applied in anomaly detection due to its generalization capability in efficiently handling non-linear data [7]. OCSVM is a variation of SVM designed to train using only positive information (normal data), in the context of unsupervised and semi-supervised learning strategies. As a kernel-based method, given training data $\mathbf{X} = \{\mathbf{x_1}, \mathbf{x_2}, \ldots, \mathbf{x_n}\}$, a compact subset of $\mathbf{R^m}$, the algorithm maps them into a higher-dimensional feature space H ($\mathbf{\Phi(x)} : \mathbf{X} \rightarrow \mathbf{H}$) via a kernel function f [8]. In this H dimension, a maximum margin hyperplane ($\mathbf{w} \cdot \mathbf{\Phi}\left(\mathbf{x_i}\right) = \rho$) is defined by the support vectors that separate the data from the origin. Hence, the training data is the first class, and the origin is the only second-class member. To achieve this separation, a quadratic programming problem must be solved [9]:

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

subject to $\omega \cdot \mathbf{\Phi}\left(\mathbf{x_i}\right) \geqslant \rho - \xi_\mathbf{i}$, $\xi_\mathbf{t} \geqslant \mathbf{0}$. Normal data are contained within the high-density region defined by the decision boundary w and equal to +1 by the function f, while the anomalies are in the sparse region and equal to -1. However, in this work, we invert these notations for clear visualization purposes: anomalies and normal data are assigned values of +1 and -1, respectively.

Vitor Dias, Marcílio Santos, Andrei Ribeiro, Fabrício R. Lobato, and João C. W. A. Costa, Institute of Technology, Federal University of Pará, Belém-Pará, Brazil, e-mail: [dias.vitor, marcilio.santos, 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

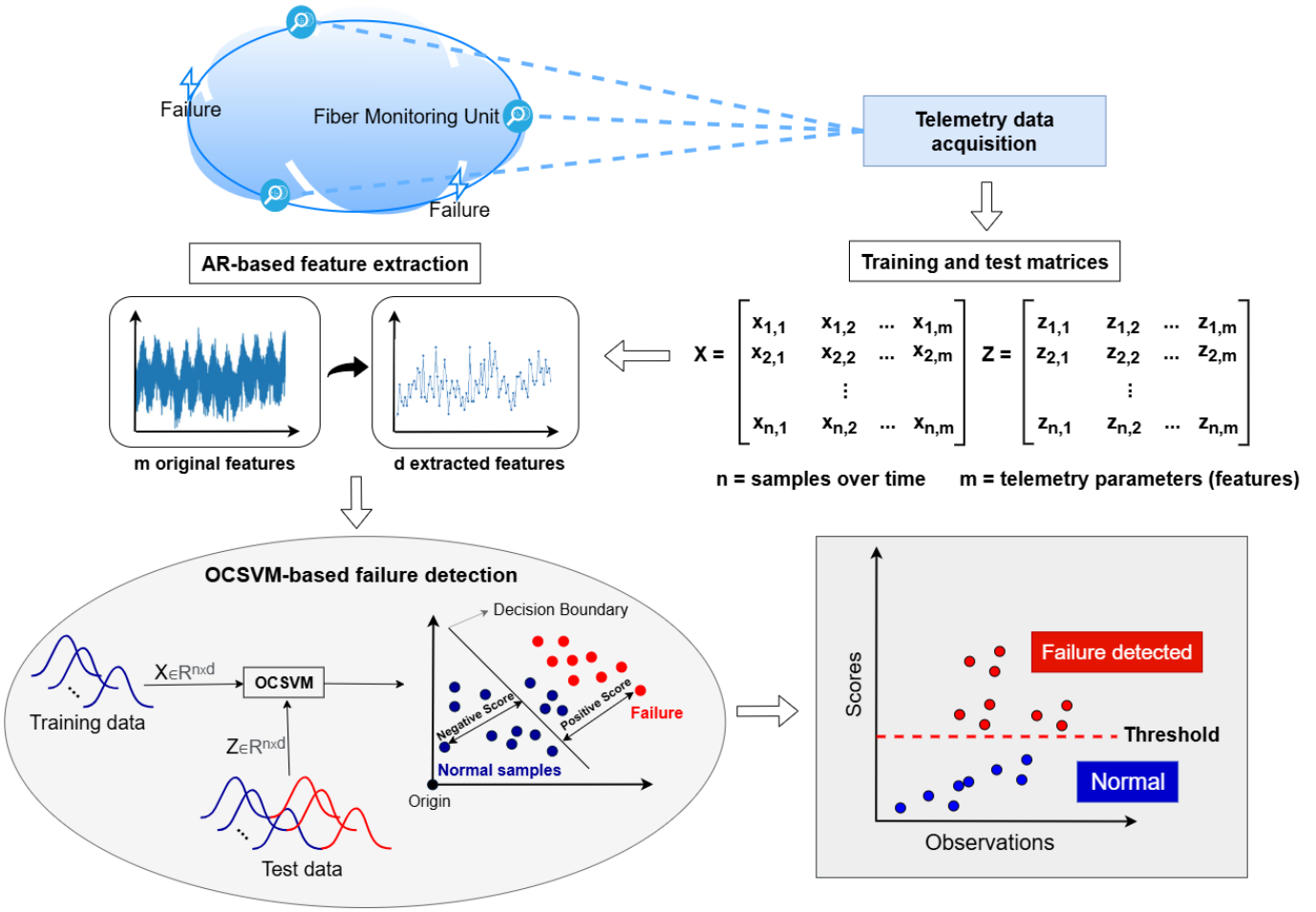


Fig. 1: Overview of the proposed approach comprising the telemetry data collection, data preprocessing, AR modeling, and OSCVM-based failure detection.

B. Autoregressive model

AR is a statistical model that predicts the next value in a time series based on a sequence of its previous values [10]. The general form of this model is given by:

$$x_t = \phi_0 + \phi_1 x_{t-1} + \phi_2 x_{t-2} + \cdots + \phi_p x_{t-p},$$

where x_t is the value of the time series at time t, which the model aims to predict, the coefficients ϕ_i represent the influence of past values x_{t-i} on the current value x_t. Finally, ϕ_0 is the white noise at time t, accounting for the portion of x_t not explained by the previous values. The order of the model, denoted by p in AR(p), indicates how many past time steps are considered in the prediction [11].

The AR feature extraction process begins with the choice of model order. The chosen autoregressive coefficients obtained after fitting the model constitute the main extracted features. Each coefficient quantifies the relationship between the value of the series at the current time and the value at one of the nine previous lags. Thus, each coefficient carries information about the strength and direction of the influence of the corresponding past value on the present value. For example, a positive and significant coefficient for the first lag (ϕ_1) indicates that the immediately previous value exerts a strong positive influence on the current value.

C. Proposed approach for failure detection

Fig. 1 depicts the proposed approach for failure detection. First, the dataset comprises samples from normal and fault conditions. Next, the training and testing data matrices are defined as X and Z, respectively.

In the training phase, only data collected under normal conditions are utilized. This data matrix comprises n samples and m features or telemetry parameters. Subsequently, the AR model is applied to each sample individually, resulting in a new dataset with d extracted features. These training features represent the temporal dependence between the values in the series and lie in a lower-dimensional space ($d < m$). In this case, the dimensionality reduction highlights the most critical variations in the data. First, the AR model is applied individually to each sample, extracting features representing the temporal dependence between the values in the series. These dimensionality reduction results aim to simplify the model and highlight the most critical variations in the data, projecting them into a lower-dimensional space. Since the data is compressed into a lower-dimensional space by the AR, the OCSVM training begins. The model calculates a decision boundary encompassing the normal data in the feature space. Thereafter, the model assigns a score corresponding to the Euclidean distance of each sample from the decision boundary that has been obtained. Finally, the threshold is generated and

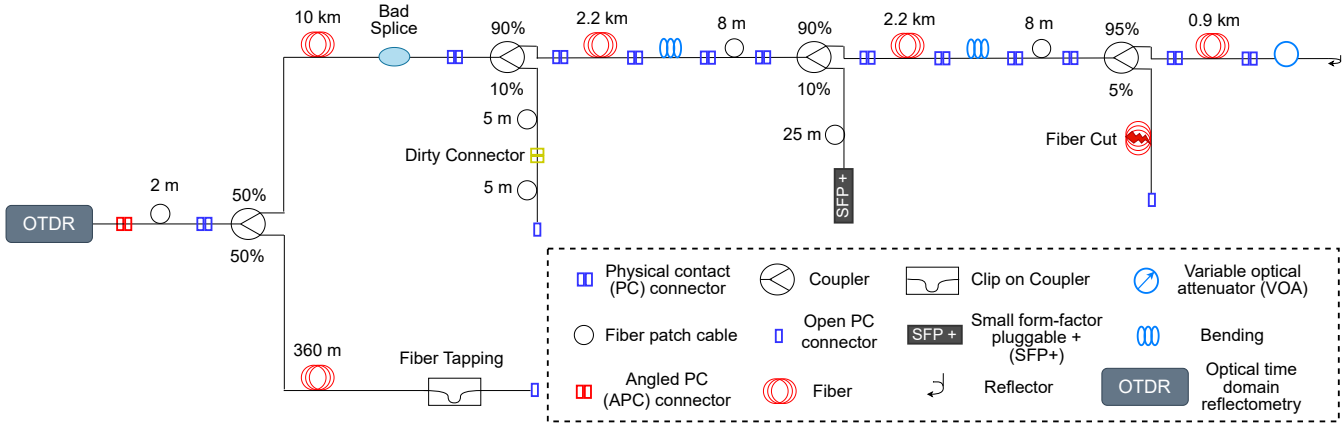


Fig. 2: Optical testbed [12].

adjusted with the parameter ν, which defines the expected proportion of anomalies within the normal data. Therefore, scores below the threshold are negative, indicating samples from normal conditions within the hyperplane.

In the testing phase, both normal and fault conditions are used. The matrix Z, with n samples and d features, represents the testing data fed into the trained OCSVM model. Therefore, for fault detection, the scores of the new data are assigned and compared with the threshold learned during training. Thus, any sample with a score above the threshold, i.e., with a positive value, is classified as a fault.

III. Results

A. Experimental setup and data acquisition

The experimental dataset, which has been made available in a public GitHub repository, is generated using a custom-designed testbed that simulates an optical network environment, as shown in Fig. 2 . The testbed is configured to reproduce real-world scenarios and emulate various types of fiber optic failures. Different fault joints are created with varying losses to produce a variable pattern of faults. To simulate different fault conditions, the testbed is designed to introduce the following failures: fiber cuts, optical eavesdropping (fiber tapping), dirty connectors, and bad splices.

The collected dataset consists of 125832 samples and 36 features. Each feature represents different optical parameters provided by the Optical Time Domain Reflectometer (OTDR), such as Signal-to-Noise Ratio (SNR). A total of 62 to 65,000 OTDR records are collected and averaged. The OTDR configuration parameters—sampling time, the pulse width and wavelength—are set to 1 ns, 10 ns, and 1650 nm, respectively. The dataset is composed of data from normal and fault conditions, presenting different types of faults, additional information is available in [12].

B. Data preprocessing and model fine tuning

Fig. 3 illustrates the dataset before AR feature extraction. It's possible to observe along the X-axis that the black and red lines intertwine, indicating that the values of these specific features are not consistently different enough to distinguish the two conditions clearly. This overlap suggests that the original features may have a low capacity for discrimination between the normal and failure states of the system being monitored. If the values of a particular feature are similar in both normal and failure conditions, that feature contributes little to the identification or classification of the problem.

Following data collection for this approach (totaling 11521 samples), the dataset was split. The first 7216 samples, corresponding to normal conditions, formed the training set to adjust the ML models. The subsequent 4305 samples constituted the test set to assess the model's performance on novel data. It is important to note that failure samples are presented only in the testing dataset. Initially, the training phase is driven by first reducing the dimensionality of the data using AR. After testing several values of p, the value with the highest training accuracy is 9.

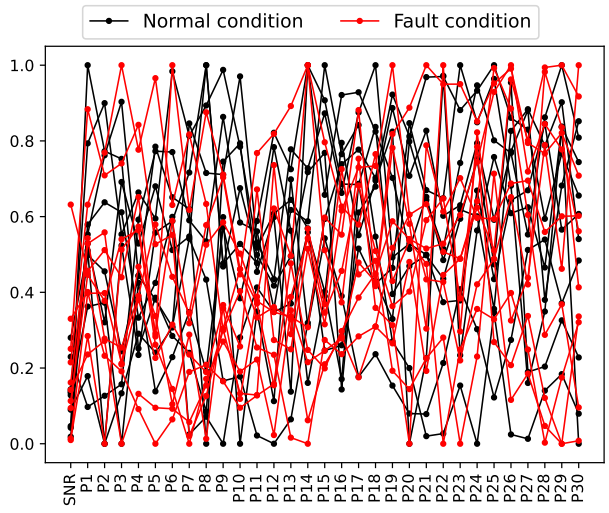


Fig. 3: Ten random samples from the training and testing sets (comprising normal and failure conditions) along with the 31 features (comprising the SNR and OTDR traces).

Therefore, the radial basis function kernel is used after fine-tuning, as it is more suitable for scenarios with high-dimensional data. The γ parameter, coefficient of the kernel function, which controls the influence of a single training sam-

ple on the shape of the decision boundary, is set to scale. Thus, γ is calculated automatically based on the number of features and their average variance. Finally, the ν parameter, used to control the proportion of failures allowed during training and the minimum fraction of support vectors, is set to the value with the highest training accuracy, 0.04. Following training, nine features are extracted from the adjusted model and stored, forming a compressed dataset in a lower-dimensional space for use in fault detection. For this reason, the integration of a dimensionality reduction method that compresses the telemetry dataset into a smaller dimensional space becomes essential to provide a scalable fault detection approach.

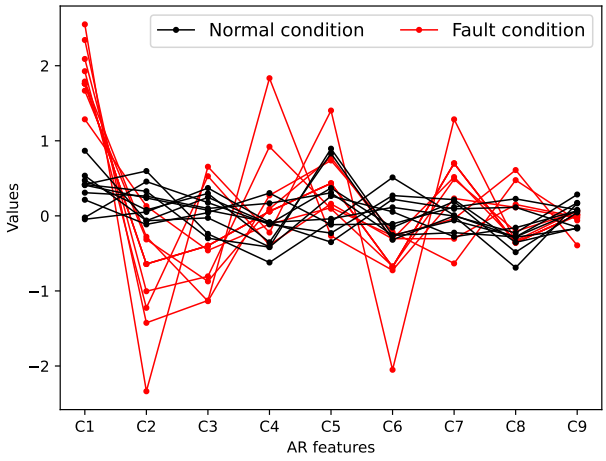


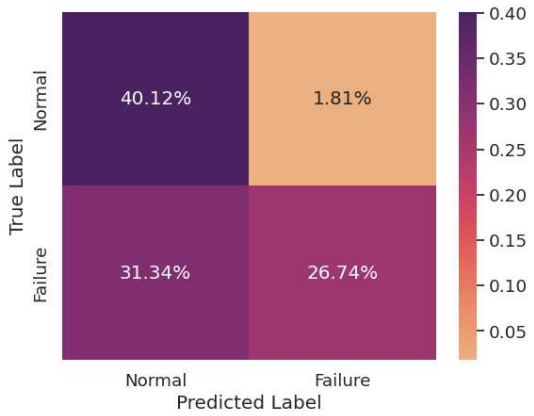
Fig. 4: Ten random samples from the training and testing sets after AR feature extraction, resulting in only 9 features.

Fig. 4 shows the new compressed dataset generated by the AR. It's possible to perceive a change in the overlap pattern between the samples in normal conditions and those with failure. Although some overlap still exists, in particular features, the red lines (failure condition) tend to exhibit more extreme positive and negative values than the black lines (normal condition), which generally remain closer to zero. This greater separation, even if not complete across all features, suggests that the AR model has managed to extract characteristics from the original data that are more discriminative between the normal and failure conditions.

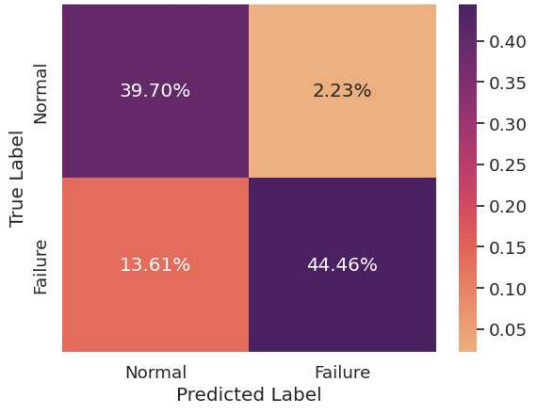
C. Fault detection results

To evaluate the performance of the proposed method, the metrics employed include accuracy and the rates of false positives (FP) and false negatives (FN). In optical networks, FP errors indicate that the model is signaling a failure even when the network is operating normally. In this sense, FP errors can lead to unnecessary maintenance actions, generating operational costs that could have been avoided. In the case of FN errors, the model does not detect a true fault in the network. FN errors can be even more damaging, leading to continuous and active faults without intervention. Therefore, finding effective ways to reduce FP and FN errors is necessary to ensure efficient resource management and network reliability.

This section evaluates two scenarios for the OCSVM-based fault detection approach: I) without feature extraction using AR, and II) incorporating AR-based features. The goal is to compare the model's performance in both configurations. Fig. 5(a) shows the performance results of OCSVM without AR feature extraction. The accuracy obtained is 66.86%, while the FN rate is 31.34%, and the FP rate is 1.81%. These results indicate that, without the dimensionality reduction provided by AR, OCSVM struggles to classify new samples and detect faults. This is evidenced by its low success rate of only 26.74% in identifying fault samples, likely due to the model's limitations when dealing with high-dimensional data. The confusion matrix in Fig. 5(b) shows the accuracy of the OCSVM with the application of AR. The model correctly identified 84.16% of the samples under failure conditions. The FN and FP rates obtained are 13.61% and 2.23%, respectively, which can be considered a low number of false alarms.



(a) OCSVM without AR feature extraction.



(b) OCSVM with AR feature extraction.

Fig. 5: Confusion matrices of the compared scenarios. The diagonal elements represent the percentage of points for which the predicted label is equal to the true label, while off-diagonal elements are those that are mislabeled by the model, i.e., the false-positive and false-negative errors.

When used without feature extraction, the OCSVM model showed limited capability in detecting faults, resulting in many FN errors. As observed in Fig. 3, most fault condition data also fall within this same range, making it challenging for the OCSVM to correctly distinguish between normal and fault behavior. This overlap in data distribution contributes to the model's difficulty in accurately identifying faults, as

fault instances exhibit characteristics similar to the normal ones. To address this limitation, features extracted using AR are introduced. As shown in Fig. 4, the AR-based features significantly improve the separability between normal and fault data. Unlike the raw input values, the AR features capture temporal patterns that are not immediately apparent, allowing the model to distinguish fault conditions better.

Moreover, Fig. 6 presents the scores obtained by the model during both the training and testing phases. As mentioned earlier, the model identify 84% of the fault samples. The black dots represent the training data used to fit the model. Some of these dots appear above the threshold line, indicating the proportion of samples the model, according to the ν parameter, expects to classify as failures. This tolerance contributes to better performance in the testing phase by reducing the number of FN errors. The blue dots correspond to normal condition data from the testing set; those above the threshold represent 2.23% of FPs. The red dots e the fault samples from the test set, and those below the threshold line account for 13.61% of FNs. These results demonstrate the effectiveness of the proposed approach in distinguishing between normal and fault conditions. Thus, combining the AR model for feature extraction with the OCSVM for anomaly detection is a strategy that contributes well to capturing temporal dependencies and identifying deviations from normal behavior, even without labeled failure data. This contribution showed satisfactory results, as FNs reduced from 31.34% to 13.61%, and accuracy increased from 66.86% to 84.17%. In this regard, the dimensionality reduction from 31 to 9 features achieved by the AR model enhances the scalability of the proposed approach.

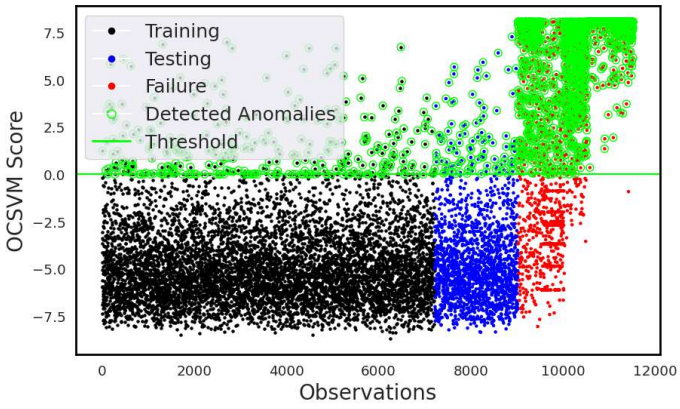


Fig. 6: Performance of the OCSVM fault detection approach over each observation.

IV. Conclusion

This work explores combining an AR-based feature extraction technique with the OCSVM model to perform fault detection in optical networks. In this sense, the proposed approach generated satisfactory results compared to the performance of OCSVM without being combined with AR. The OCSVM without AR achieved an accuracy of 66.86%, as well as 1.81% and 31.34% FPs and FNs, respectively. Therefore, a considerable performance gain was observed when using OCSVM in combination with AR, achieving an accuracy of 84.16%, as well as 2.23% and 13.61% FPs and FNs, respectively. The proposed approach adequately overcomes the imbalance in the data, as it proved capable of efficiently detecting faults. In addition, the reduction in FN errors resulted in a 17% increase in accuracy.

Acknowledgments

This work was supported by the National Council for Scientific and Technological Development (CNPq) and Amazon Sustainability Institute with Science and Innovation (iSACI).

References

[1] F. Musumeci, C. Rottondi, G. Corani, S. Shahkarami, F. Cugini e M. Tornatore. “A tutorial on machine learning for failure management in optical networks,” *Journal of Lightwave Technology*, vol. 37, no. 16, pp. 4125–4139, 2019.

[2] A. Binbusayyis e T. Vaiyapuri. “Unsupervised deep learning approach for network intrusion detection combining convolutional autoencoder and one-class SVM,” *Applied Intelligence*, vol. 51, no. 10, pp. 7094–7108, 2021.

[3] D. Wang, C. Zhang, W. Chen, H. Yang, M. Zhang e A. P. T. Lau. “A review of machine learning-based failure management in optical networks,” *Science China Information Sciences*, vol. 65, no. 11, p. 211302, 2022.

[4] J. M. Pedro e N. Costa, “Model and data-centric machine learning algorithms to address data scarcity for failure identification,” *Journal of Optical Communications and Networking*, vol. 16, no. 3, pp. 369–381, 2024.

[5] M. Furdek, C. Natalino, A. Di Giglio e M. Schiano. “Optical network security management: requirements, architecture, and efficient machine learning models for detection of evolving threats,” *Journal of Optical Communications and Networking*, vol. 13, no. 2, pp. A144–A155, 2021.

[6] S. K. Patri, I. Dick, K. Kaeval, J. Müller, J.-J. Pedreno-Manresa, A. Autenrieth, J.-P. Elbers, M. Tikas e C. Mas-Machuca, “Machine learning enabled fault-detection algorithms for optical spectrum-as-a-service users,” *Proceedings of the International Conference on Optical Network Design and Modeling (ONDM)*, pp. 1–6, 2023.

[7] B. Schölkopf e A. J. Smola, “Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond” . Cambridge, MA, EUA: MIT Press, 2002.

[8] B. Schölkopf, A. Smola e K.-R. Müller. “Nonlinear Component Analysis as a Kernel Eigenvalue Problem,” *Neural Computation*, vol. 10, no. 5, pp. 1299–1319, 1998.

[9] B. Schölkopf, J. C. Platt, J. Shawe-Taylor, A. J. Smola e R. C. Williamson. “Estimating the support of a high-dimensional distribution,” *Neural Computation*, vol. 13, no. 7, pp. 1443–1471, 2001.

[10] M. Dalal, A. C. Li e R. Taori. “Autoregressive models: What are they good for?” arXiv preprint arXiv:1910.07737, 2019.

[11] S. Bond-Taylor, A. Leach, Y. Long e C. G. Willcocks. “Deep generative modelling: A comparative review of vaes, gans, normalizing flows, energy-based and autoregressive models,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 44, no. 11, pp. 7327–7347, 2021.

[12] K. Abdelli, J. Y. Cho, F. Azendorf, H. Griesser, C. Tropschug e S. Pachnicke. “Machine-learning-based anomaly detection in optical fiber monitoring,” *Journal of Optical Communications and Networking*, vol. 14, no. 5, pp. 365–375, 2022.