Experimental Analysis of Adaptive ML Classifiers for Dynamic Detection of Emerging Physical-Layer Attacks

Aleksandra Knapińska[1, 2] and Marija Furdek[1]

[1] Chalmers University of Technology, Gothenburg, Sweden, {alekna, furdek}@chalmers.se
[2] Wrocław University of Science and Technology, Wrocław, Poland

Abstract *We experimentally evaluate three ML classifiers for detecting physical-layer attacks in optical networks. Using telemetry data from a real-world testbed under six attack types, we achieve Balanced Accuracy of up to 0.936 for unseen attacks, and demonstrate rapid adaptation of Multilayer Perceptron to evolving threats. ©2025 The Author(s)*

Introduction

The key role of optical networks as critical communication infrastructure makes their security and resilience fundamental to reliable and secure global communications. The significance of optical networks marks them as enticing targets of attacks aimed at disrupting services of eavesdropping communication. Service disruption attacks include insertion of jamming signals[1] and polarization scrambling attacks[2]. In in-band jamming, the wavelength of the harmful signal is the same as the targeted optical channel, adding unfilterable noise. In out-of-band jamming, the high-powered inserted signal is at a different wavelength and reduces the limited amount of erbium-doped fiber amplifier (EDFA) gain, resulting in insufficient compensation of span loss. In a polarization scrambling attack, the fiber is squeezed at a high frequency, causing fluctuations in the state of polarization that are too fast for the coherent receiver to compensate for, resulting in burst errors.

Tampering with only a few optical network links or nodes can cause a widespread disruption of the aggregate carried services, with rippling effects to a multitude of overlay network services. For example, the recent Red Sea cables cuts disrupted 25%-70% of the Europe-Asia traffic flows[3], while in the German railway attack, targeted cuts of only two optical fibers caused major railway traffic halts[4]. Various authorities[5],[6] call for action against evolving network attacks that increase in frequency, size, and sophistication. To cope with the evolving network security threat landscape, quick and accurate detection of attacks is critical to fast and effective network recovery. Machine learning has been proven to be a particularly successful tool for attack detection, capable of identifying subtle changes in optical performance monitoring (OPM) parameters and attributing them to different physical-layer attack techniques[2],[7].

Traditionally, two main ML approaches have been applied to physical layer security: specialized detectors based on supervised learning, designed to detect specific attack types[8],[9], and unsupervised learning models that identify anomalies in telemetry data[10],[11]. Specialized detectors are characterized by extremely high accuracy, as they are tailored to specific patterns associated with particular attacks. However, they require extensive training datasets with a substantial number of samples for each specific attack. Furthermore, they operate on a finite set of possible attacks; if a new attack type emerges, it may go undetected due to the absence of a corresponding detector and training data. Unsupervised models, on the other hand, do not require labeled data and can be integrated into the general pipeline using aggregated telemetry data. They can detect a wide range of anomalies, but cannot provide fine-granular diagnostic information and often lack adaptability and mechanisms to incorporate feedback on their prediction performance for continuous improvement.

In this paper, we address these challenges and investigate the need for specialized ML attack detectors in detection of emerging physical-layer attacks. We consider three different attack detection scenarios: *i)* detection of previously known and trained for attack types, *ii)* detection of unseen attacks upon training on other attack types, and *iii)* detection of dynamically emerging, unseen attacks. Using a real-world experimental dataset comprising six physical-layer attacks, we evaluate the performance of three classifiers from distinct model families: the neural-network-based Multilayer Perceptron (MLP), the tree-based eXtreme Gradient Boosting (XGB), and the kernel-based Support Vector Machine (SVM). Our findings indicate excellent performance in detecting previously unseen attacks, achieving Balanced ACcuracy (BAC) score values of up to 0.936. Furthermore, we explore dynamic model update strategies to handle evolving threats, with simultaneous and sequential occurrences of unseen attacks by leveraging the *partial fitting* capabilities of MLP. This enables rapid adaptation to evolving conditions, quickly converging to BAC values over 0.9.

Experimental Environment

In this study, we use the dataset from[2], obtained from an operator testbed and comprising normal traffic alongside six types of attacks. These attacks fall into three main categories: in-band jamming (INB), out-of-band jamming (OOB), and polarization scrambling (POL), each performed at lighter (LGT) and stronger (STR) intensity. Accordingly, the six attack types are: INBLGT, INBSTR, OOBLGT, OOBSTR, POLLGT, and POLSTR.

The OPM samples characterizing each condition are collected from an experimental optical network testbed with coherent transceivers, Reconfigurable Optical Add-Drop Multiplexers (ROADMs) and EDFAs. A detailed description of the experiments and testbed can be found in our previous work[2]. The monitored channels are two optical 200 Gbit/s polarization multiplexed 16 quadrature amplitude modulation (16QAM) signals at 195.2 and 195.3 THz.

The data consists of measurements of 31 different OPM parameters. The dataset includes 1440 one-day traces for each attack type, as well as for normal traffic. To simulate a realistic, imbalanced scenario, each experiment involved random sampling of normal and attack traffic to create an imbalanced class distribution of 100:15 (for every 100 *normal* samples there are 15 *attack* ones). To ensure statistical significance, all experiments were repeated 100 times and the average values are reported.

We used the `scikit-learn`[12] implementations of MLP (one hidden layer of 100 neurons, *ReLU* activation, and *adam* optimizer) and SVM (*rbf* kernel, C = 1.0). We used the authors' implementation of XGB[13]. The dynamic classification of evolving attacks was implemented using the *test-then-train* protocol implemented in `stream-learn`[14].

We evaluate the performance of the classifiers using the Balanced ACcuracy (BAC) score as the metric, defined as the average of recall obtained on each class, which indicates the ability of a model to detect all instances of attacks while avoiding false alarms. In all cases, the task is binary classification, where the model determines whether a given sample is an attack or not.

Attack-Tailored Detectors

The first experiment evaluates the detection performance of specialized classifiers trained and tested on individual attack classes. We also include an aggregated classifier trained on anonymized samples from all six attacks (referred to as the *mixed* scenario). It maintains the 100:15 class distribution, but with significantly greater diversity in the minority class, as it includes samples from all attack types. For each experiment run, we use a dataset of 2300 samples, with 80% allocated for training and the remaining 20% for testing.

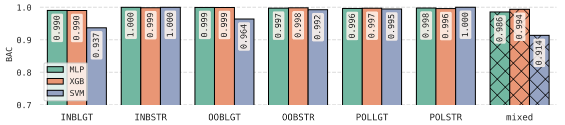

Fig. 1: BAC for the dedicated and the mixed attack detectors.

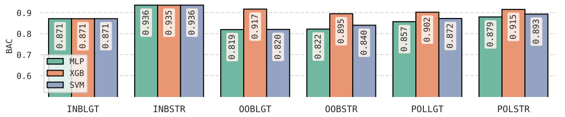

Fig. 2: BAC when detecting previously unseen attacks.

Fig. 1 shows almost perfect attack classification by all considered classifiers, achieving BAC values between 0.937 and 1. Remarkably, the performance for the *mixed* scenario drops only slightly to 0.914-0.994. Among the classifiers, MLP and XGB perform consistently well, while SVM has slightly more difficulties generalizing in the *mixed* attack scenario. These results demonstrate that training classifiers on mixed samples from different attacks is sufficient to learn what constitutes an attack and to detect it successfully.

Detection of Unseen Attacks

As malicious actors continuously evolve their techniques, we cannot always count on the presence of all attack types in the training set. Therefore, in this experiment, we evaluate the classifiers' ability to detect previously unseen attack types. Therefore, the minority class comprises five different anonymized attack types during training, and the remaining unseen attack type during testing.

Fig. 2 shows the BAC scores of classifiers trained on five attack types and detecting the sixth. The overall prediction quality is remarkable, with BAC scores of 0.819-0.936 for all classifiers. The INBSTR attack appears as the easiest to detect when unseen, with the highest BAC value for all classifiers. The OOBLGT attack emerges as the most challenging, although XGB has no difficulty detecting it. Among the classifiers, XGB generalizes best across all scenarios, likely due to its ability to capture benign traffic patterns or identify commonalities across different attacks. These results further demonstrate that successful detection of each attack type does not require its inclusion during training. Interestingly, SVM, that performed the poorest in the previous experiment, exhibits solid generalization capabilities in this scenario, highlighting its potential for detecting previously unseen attacks. These results indicate that combining multiple attack scenarios into a joint training set enables the detection of new intrusions, ultimately contributing to improved network security.

Dynamic Attack Detection by the MLP

In addition to the need for robust and well-generalizing models, they need to continuously upgrade with newly acquired knowledge from the

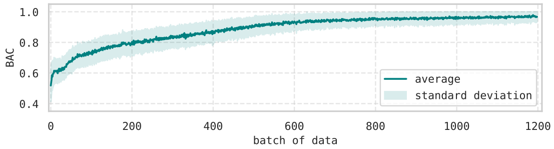

Fig. 3: BAC for the dynamically updated attack-type-agnostic MLP model with simultaneous occurrences of different attacks.

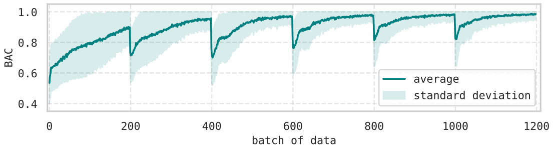

Fig. 4: BAC for the dynamically updated attack-type-agnostic MLP model with sequential occurrences of unseen attacks.

dynamically changing network ecosystem, creating an adaptive feedback loop. To explore this, the final experiment introduces a dynamic scenario in which we leverage the partial-fitting capabilities of the MLP classifier to periodically update the model on the continuously incoming data.

To this end, we interpret the telemetry information as a data stream and process it sequentially. Initially, the model is trained on a small batch of 100 samples and classifies each incoming sample. Every 100 samples, the model is updated with their corresponding labels using the *partial fit* method. Thus, the model is not reset; instead, its internal parameters (e.g., neuron weights) are updated with the new data, while retaining previously learned knowledge. As a result, there is no need to store the previously processed data, as its characteristics are embedded in the model. The batches maintain the 100:15 imbalanced class distribution. This setup maintains a continuous and lightweight feedback loop, enabling the model to retain previously learned knowledge while adapting to newly observed data. As a result, the processing time for subsequent batches remains constant.

We first adapt the *mixed* scenario from our initial experiment, where all attacks are considered jointly. Fig. 3 shows the BAC achieved by the MLP classifier over 1200 consecutive data batches. Despite the constant arrival of previously unseen samples, prediction quality consistently improves over time. As more data become available during the network lifetime, the classifier continues to learn and improve its performance. A stable and satisfactory BAC level of 0.9 is reached after processing approximately 500 batches. Beyond that point, the model continues to refine its detection capabilities, making attacks increasingly less surprising.

Remarkably, processing a single data batch, including both prediction and model update, takes only 5.83 milliseconds on average when run on a standard laptop. Therefore, the time required to keep such a model continuously up-to-date is almost negligible, further highlighting the high practical applicability of the proposed scheme.

Next, we examine how the continuously updated detector responds to new attack types introduced sequentially. To this end, while maintaining the 100:15 class distribution, we inject attack samples from only one type at a time. Specifically, the first 200 batches contain a single attack type, the next 200 batches a different one, and so on. All attack types share the same label, so the model is unaware that the nature of the attacks is changing over time. Since our dataset includes six attack types, there are 720 possible attack orderings. In the following part, we analyze the results averaged over 100 randomly selected sequences.

Fig. 4 shows the BAC over time in this scenario. We can clearly observe a drop in detection quality whenever a new attack type is introduced, followed by a rapid recovery as the model adapts. Remarkably, with each new attack, the drop in performance becomes smaller, and the return to near-perfect accuracy happens more quickly.

From a practical perspective, employing a dynamically updating binary attack detection model offers an effective solution to evolving threats. Such a model is not only capable of recognizing new intruder techniques but also of adapting and learning continuously during network operation. It is also important to emphasize that the models evaluated in our study require only small amounts of data, demonstrating that high prediction quality can be achieved without significant computational overhead.

Conclusions

In this paper, we addressed the problem of physical-layer attack detection from multiple perspectives, examining the need for specialized attack detectors. Using a real-world dataset containing six distinct attack types, we demonstrated that various classifiers can effectively generalize across them and reliably raise alarms when all attacks are treated jointly within a binary classification framework. Moreover, we showed that previously unseen attacks can also be detected with strong accuracy, even without being present in the training data. These results suggest that deploying dedicated models for each attack type is not necessary, as joint, traffic-type-agnostic models perform remarkably well and are capable of detecting novel attacks with solid effectiveness.

Furthermore, we explored dynamic scenarios in which attacks gradually emerge as intruders refine their techniques over time. Our experiments demonstrated that continuously updated classifiers can quickly learn and adapt, successfully detecting previously unseen attacks by incrementally incorporating knowledge from small batches of new data.

Acknowledgments

This work was supported by the Swedish Research Council (2023–05249), and the European Commission's Digital Europe Programme (101127973) through the 5G-TACTIC project. The authors gratefully acknowledge Marco Schiano and Andrea Di Giglio from Telecom Italia for their support in gathering experimental data, and Infinera for providing the Groove G30 transponders used in the experiments.

References

[1] N. Skorin-Kapov, M. Furdek, S. Zsigmond, and L. Wosinska, "Physical-layer security in evolving optical networks", *IEEE Communications Magazine*, vol. 54, no. 8, pp. 110–117, 2016. DOI: 10.1109/MCOM.2016.7537185.

[2] M. Furdek, C. Natalino, F. Lipp, D. Hock, A. Di Giglio, and M. Schiano, "Machine learning for optical network security monitoring: A practical perspective", *Journal of Lightwave Technology*, vol. 38, no. 11, pp. 2860–2871, 2020. DOI: 10.1109/JLT.2020.2987032.

[3] telecoms.com (Oct. 2024): "*Red Sea cable cuts' impact was 'severely underestimated'*" (www.telecoms.com/telecoms-infrastructure/red-sea-cable-cuts-impact-was-severely-underestimated-).

[4] bbc.com (Nov. 2024): "*Germany suspects sabotage behind severed undersea cables*" (www.bbc.com/news/articles/c9dl4vxw501o).

[5] EU Critical Entities Resilience (CER) Directive, 2023 (tinyurl.com/2kevhdnn).

[6] US National Cybersecurity Strategy, 2023 (tinyurl.com/hfk278hd).

[7] C. Natalino, M. Schiano, A. Di Giglio, and M. Furdek, "Root cause analysis for autonomous optical network security management", *IEEE Transactions on Network and Service Management*, vol. 19, no. 3, pp. 2702–2713, 2022. DOI: 10.1109/TNSM.2022.3198139.

[8] J. Sakhnini, H. Karimipour, A. Dehghantanha, and R. M. Parizi, "Physical layer attack identification and localization in cyber–physical grid: An ensemble deep learning based approach", *Physical Communication*, vol. 47, p. 101 394, 2021. DOI: 10.1016/j.phycom.2021.101394.

[9] T. M. Hoang, T. Q. Duong, H. D. Tuan, S. Lambotharan, and L. Hanzo, "Physical layer security: Detection of active eavesdropping attacks by support vector machines", *IEEE Access*, vol. 9, pp. 31 595–31 607, 2021. DOI: 10.1109/ACCESS.2021.3059648.

[10] C. Natalino, C. Manso, L. Gifre, *et al.*, "Microservice-based unsupervised anomaly detection loop for optical networks", in *Optical Fiber Communication Conference*, 2022, Th3D–4. DOI: 10.1364/OFC.2022.Th3D.4.

[11] P. Lechowicz, C. Natalino, V. Karunakaran, A. Autenrieth, T. Bauschert, and P. Monti, "Trade-offs in implementing unsupervised anomaly detection with tapi-based streaming telemetry", in *25th International Conference on High Performance Switching and Routing (HPSR)*, 2024, pp. 13–18. DOI: 10.1109/HPSR62440.2024.10635922.

[12] F. Pedregosa, G. Varoquaux, A. Gramfort, *et al.*, "Scikit-learn: Machine learning in Python", *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

[13] T. Chen and C. Guestrin, "XGBoost: A scalable tree boosting system", in *22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2016, pp. 785–794.

[14] P. Ksieniewicz and P. Zyblewski, "Stream-learn—open-source python library for difficult data stream batch analysis", *Neurocomputing*, vol. 478, pp. 11–21, 2022.