Optical Network Security Management: Requirements, Architecture and Efficient Machine Learning Models for Detection of Evolving Threats

Citation for the original published paper (version of record):
Furdek Prekratic, M., Natalino Da Silva, C., Giglio, A. et al (2021). Optical Network Security Management: Requirements, Architecture and Efficient Machine Learning Models for Detection of Evolving Threats [Invited]. Journal of Optical Communications and Networking, 13(2): A144-A155. http://dx.doi.org/10.1364/JOCN.402884

Optical Network Security Management: Requirements, Architecture and Efficient Machine Learning Models for Detection of Evolving Threats [Invited]

Marija Furdek[1,*], Carlos Natalino[1], Andrea Di Giglio[2], and Marco Schiano[2]

[1] Department of Electrical Engineering, Chalmers University of Technology, 412 96 Gothenburg, Sweden
[2] Telecom Italia, Via Guglielmo Reiss Romoli, 274 – 10148 Turin, Italy
[*] Corresponding author: furdek@chalmers.se
[*] The implementation of this work is available at https://github.com/carlosnatalino/2020-JOCN-efficient-ML

Compiled November 3, 2020

As the communication infrastructure that sustains critical societal services, optical networks need to function in a secure and agile way. Thus, cognitive and automated security management functionalities are needed, fueled by the proliferating Machine Learning (ML) techniques and compatible with common network control entities and procedures. Automated management of optical network security requires advancements both in terms of performance and efficiency of ML approaches for security diagnostics, as well as novel management architectures and functionalities. This paper tackles these challenges by proposing a novel functional block called Security Operation Center (SOC), describing its architecture, specifying key requirements on the supported functionalities and providing guidelines on its integration with optical layer controller. Moreover, to boost efficiency of ML-based security diagnostic techniques when processing high-dimensional optical performance monitoring data in the presence of previously unseen physical-layer attacks, we combine unsupervised and semi-supervised learning techniques with three different dimensionality reduction methods and analyze the resulting performance and trade-offs between ML accuracy and run time complexity.

https://doi.org/10.1364/JOCN.402884

1. INTRODUCTION

Uninterrupted and secure operation of high-performance communication networks is one of the key enablers of the ongoing evolution towards a networked, information society. Correspondingly, critical network infrastructure proves as an attractive target of counterfeit attacks aimed at breaching communication confidentiality, integrity or availability. Optical networks, as the only future-proof solution capable of supporting the proliferating bandwidth-hungry applications, are no exception from the expanding landscape of security threats which evolve in their sophistication and scale. Attacks targeting the optical layer can, according to their objective, roughly be divided into eavesdropping and service disruption. Eavesdropping is aimed at gaining unauthorized access to the information transmitted through the optical fiber (e.g., by abusing a well-known monitoring method of bending the fiber and creating temporary couplers [1]), and can be counteracted by encryption at different network layers [2].

Service disruption attacks, which are in the focus of this work, can be performed by methods of varying levels of sophistication, scope, persistence, or effects. Physical breaches into the network infrastructure can be exploited to sever the fibers, causing outright service interruption, to insert harmful signals (e.g., jamming, which exacerbates non-linear impairments) or to negatively affect fiber properties (e.g. fiber squeezing that leads to error-inducing fast variations in the polarization state of light). Effects of physical-layer breaches can propagate to upper layer services, resulting in correlated cascading failures. Attacks can be designed to evade detection, or maximize the damage by adapting to known counteractions, making network security management a complex challenge.

To cope with the ever-evolving intelligent adversary, cybersecurity is strongly benefiting from Artificial Intelligence (AI)-driven automation across all network domains and applications [3]. In general, AI-based approaches are finding a widening application in a variety of tasks related to cognitive and automated optical network management and control. As a viable alternative to the ineffective and unreliable use of predetermined thresholds on network performance indicator values for invoking network adaptation [4], AI techniques, and in particular ML approaches as their subset, are considered pivotal for realizing the Collect-Analyze-Test loop for autonomous network operation [5, 6]. Examples of successful applications of

ML to complex optical networking problems include Quality of Transmission (QoT) estimation [7], nonlinearity monitoring and launch power optimization [8], as well as management of various faults (overviewed in [9]) such as filter shift and tightening [10], or unexpected signal power reduction [11]. These approaches predominately rely on Supervised Learning (SL) techniques, e.g. Artificial Neural Networks (ANNs) or Support Vector Machines (SVMs), trained on *a priori* labeled data that contains complete information about the mapping between the input given to the models and the expected output.

Since physical-layer attacks cause intricate changes in optical signal parameters which vary drastically across different attack techniques, defining exact analytical models or thresholds for triggering countermeasures has not been proven viable towards cognitive physical-layer security management. Instead, strong potential of ML techniques to support this cause has been demonstrated in several ways. In [12], SVM was applied to detect the presence of unauthorized optical signals by inspecting the optical spectrum. ANN-based approach for detecting high-power jamming attacks was proposed in [13]. Detection and identification of attacks using experimental data obtained by performing in- and out-of-band-jamming, as well as polarization scrambling attacks at the link level was carried out in [14] using various SL techniques. In addition to SL, the work in [15] focused on the application of Unsupervised Learning (UL) and Semi-Supervised Learning (SSL) techniques to network-level detection of attacks. As UL and SSL are not trained on previously collected, labeled Optical Performance Monitoring (OPM) data, they can detect even novel, previously unseen attack techniques, making them a worthwhile contender for coping with the evolving threat landscape.

In spite of the tremendous breakthroughs and continuous refinements in capabilities and performance of ML techniques, their integration into production environments is still at its infancy due to several important challenges. The first set of challenges stems from practical system integration and interoperability issues. Run-time carrier-grade deployment of ML-based techniques in Network Management Systems (NMSs) requires a framework for autonomous optical network security management tightly integrated with existing workflows and tools. The ML models need to be accessible to a variety of network elements, ranging from optical nodes to multi-domain orchestrators, in order to enable multi-domain security management, as well as federated or hierarchical learning [16]. ML models should feature multi-protocol adaptive interfaces exposing their functionalities to the network elements involved in security management routines, supporting input formats inherent to both current and legacy devices that provide machine-readable OPM and/or visual channel representation data. In addition, ML models should execute fast to allow incorporation into carrier-grade interval-defined monitoring cycles during which the OPM data must be collected, analyses by the ML models must be carried out, their outputs consolidated in the Software Defined Networking (SDN) controller, and required network-level actions must be triggered. To support cognitive network operation with monitoring cycles expected to tighten, low-complexity (training and/or inference) models used in conjunction with purpose-specific ML accelerators, containerization and load balancing are key to the implementation of encompassing security management without impacting control procedures in place.

The second set of challenges is related to the performance of ML models themselves. To ensure correct and reliable operation, high model accuracy is the primary requirement. However, in reality different models obtain different false positive and false negative rates, so their applicability greatly depends on the use case. For example, from a security point of view, an operator might be willing to trade off slightly higher false positive rates to certain threats, which can impact the resource usage efficiency due to triggering unnecessary reconfigurations, for zero false negative rates, thus ensuring that attacks do not go unnoticed. Accuracy of deployed ML models can also be improved by using a sliding window approach, where the impact of inaccuracies detected over an observation window can be leveled out by carefully dimensioning the window, or by using ensemble and symbolic models that intertwine different ML models and/or specialist knowledge. The ML models outputs should be interpretable to support the security analysts keeping pace with the sheer volume of alerts being generated [17].

Finally, consolidation of coherent transceivers as the *de facto* next generation optical transmission technology [18] enables the collection of a rich set of OPM data with tens of features. In Deep Learning (DL) models, which have built-in feature extraction layers, this high number of features has the potential to positively impact the model accuracy, while features which do not contribute to the accuracy can be filtered out. However, SSL and UL models do not have built-in feature extraction capabilities. This means that an increased number of features may impact negatively on the accuracy of the models [19]. In such cases, dimensionality reduction methods can be applied in the dataset pre-processing phase, removing the features with lower relevance to the problem [20]. The potential benefits of combining dimensionality reduction with SSL and UL are twofold: *(i)* the accuracy of the SSL/UL model can be improved by extracting only relevant data and *(ii)* the run time of the SSL/UL model can be reduced, since less data needs to be processed.

In [21], we identified some of the challenges related to achieving carrier-grade performance of ML techniques for optical-layer security management. This paper extends the high-level considerations from [21] by proposing guidelines for incorporating security assurance into the optical network management architecture. To this end, we propose a new functional block which we call the Optical Security Manager (OSM), describe its architecture and define key requirements on the OSM functionalities. We then carry out an experimental case study of SSL and UL techniques for physical-layer attack detection. To improve efficiency of the approaches, we apply different dimensionality reduction methods to the SSL and UL models for the attack detection task and analyze the obtained trade-offs between performance and scalability.

The remainder of the paper is organized as follows. Section 2 presents the security-oriented network management architecture and describes the main Network Security Manager features and requirements. Section 3 details the background of dimensionality reduction methods applied to un- and semi-supervised ML techniques, followed by a detailed performance analysis in Section 4. Several open challenges related to optical network security management are summarized in Section 5, and concluding remarks are given in Section 6.

2. OPTICAL LAYER SECURITY IN EVOLVING NETWORK OPERATION

A. Network Security Management Framework

Network security management in general relies on three main pillars, illustrated in Fig. 1 [21]. *Risk management* entails the development of accurate risk models capturing the versatile effects

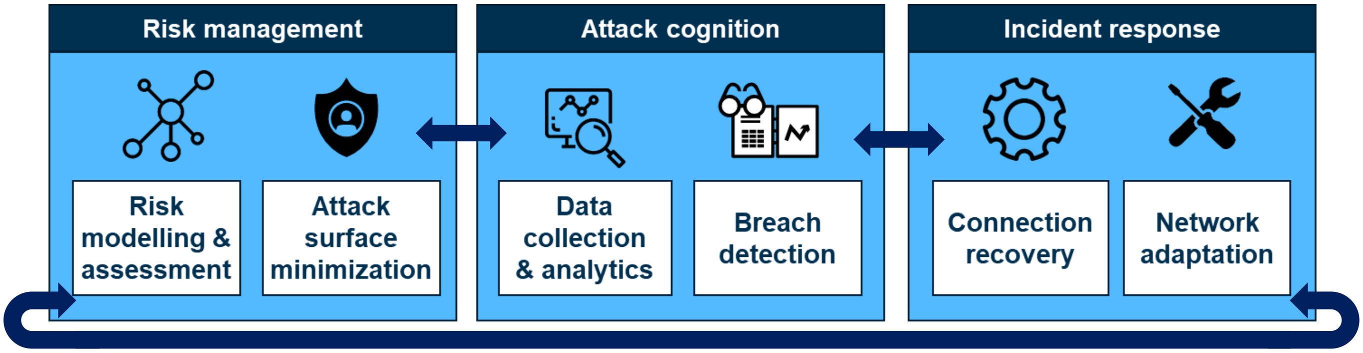


Fig. 1. High-level view of a network security management loop.

of different attack vectors and gauging the impact of these effects on the network. Based on such models, this pillar also includes minimization of the network surface exposed to attacks. Guaranteeing a certain level of robustness to known and emerging security threats requires continuous updates of the risk models and security-enhancing network design. Examples of model instances for gauging the effects of targeted fiber cut attacks on the optical network can be found in [22], while examples of attack-aware optical network planning under static, periodic and dynamic traffic can be found in [23], [24] and [25].

Attack cognition encompasses the gathering of representative and indicative network performance data, performing its deep analysis, and correctly attributing the observed trends to different types of security breaches in order to detect an attack and identify the location of the breach. This requires detailed knowledge about possible attack entry points and the so-called signatures of different attack methods, i.e., their effects on the distinct OPM parameters. Due to the sparse deployment of costly OPM equipment, ubiquitous, real-time collection of OPM data in optical networks is challenging. However, the latest generation of coherent receivers with rich Digital Signal Processing (DSP) functionalities alleviates this issue by collecting a rich OPM dataset at the destination of each connection (e.g., on a per-minute basis) and exposing it to the NMS through standardized interfaces. In addition to the examples of approaches for detecting the presence of physical-layer breaches summarized in Sec. 1, an approach for localization of high-power jamming signals can be found in [26], while [15, 27] describe a framework for attack monitoring probe design to aid localization of harmful connections and/or breached links.

Based on the security diagnostics, the *incident response* pillar entails recovery of affected network elements and connections, neutralization of the breach and network adaptation to improve resilience towards potential future occurrences of similar attacks. Network adaptation can benefit from e.g. attack-aware preplanning of backup resources [28], fast frequency hopping [29], connection rerouting, modulation format and spectrum reassignment (e.g., using a procedure described in [30]), or periodical proactive resource reallocation [13].

B. Security Assurance in Optical Network Management Architecture

Optical Security Assurance (OSA) must be tackled in the framework of Cognitive Network Management System (C-NMS) evolving architectures [4]. The advent of Transport-SDN concept and the development of open source multi-vendor network controllers have deeply transformed the optical network management vision [31]. The crucial aspects of the new C-NMS paradigm can be summarized as follows.

- The Network Management and Control is a cross-layer comprehensive architecture composed by a number of network controllers (each one dedicated to a specific network layer or operation domain) and one or more orchestrators;
- Each controller provides end-to-end services that are set-up on demand by an upper layer orchestrator;
- Standard Application Programming Interfaces (APIs) provide easy access to network services for users and external systems (e.g. Data Center Hypervisors);
- Analytics are widely used to effectively diagnose network status and implement the appropriate countermeasures for any malfunction.

Security assurance represents a new important feature of optical transport services that must be smoothly introduced in the present Transport-SDN control architecture. Another crucial aspect of optical security management is the need for its tight integration in the processes of the SOC, the organization in charge of the whole network and Information Technology security in many large companies. In other words, OSA lays on the borderline between network and security management. Therefore, any conflict or superposition of roles and responsibilities between the Network Operation Center (NOC) and the SOC must be absolutely avoided. For these reasons, the introduction of security assurance in optical networks is a non-trivial task with the following essential functionalities and requirements:

- telemetry (if not provided by the optical controller);
- attack detection and classification;
- prompt attack reaction for service downtime minimization;
- attack localization;
- permanent attack remediation.

C. Optical Security Manager (OSM) Architecture

The functions described in the previous subsection must cooperate smoothly with the optical layer controller, the Transport-SDN component that manages all optical layer functions and exposes end-to-end optical transport services to a network orchestrator. Rather than redesigning the architecture of optical layer controllers which is well established in the open source community

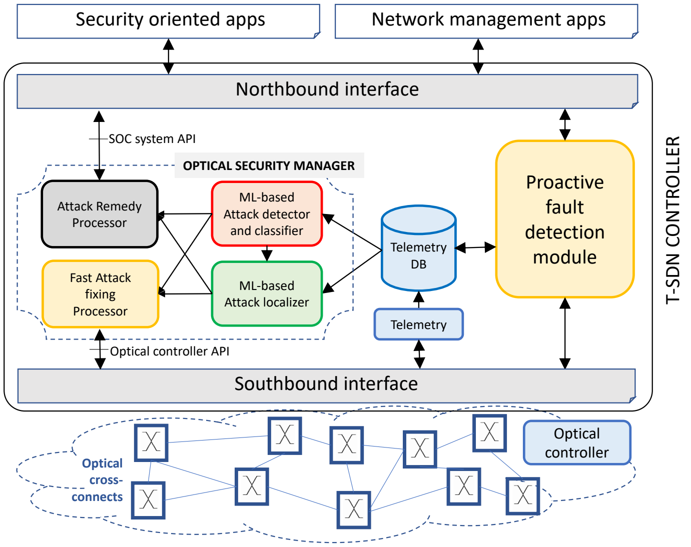


Fig. 2. The proposed OSM architecture in the framework of a cognitive Transport Software Defined Network (T-SDN) controller (adapted from [4]).

(see for instance [32]), we propose to complement the optical layer controller functions with the introduction of an OSM as a new functional block. The OSM architecture, shown in Fig. 2, encompasses all the optical security functions and is in principle compatible with any optical controller. It also provides a suitable interface towards SOC systems.

The OSM is composed by a *Telemetry* and *Telemetry data base (DB)* blocks that retrieve and store the coherent transceiver OPM data (and possibly other relevant network status data) [15]. The OPM data are used by the two ML modules: the *Attack Detector and Classifier* and the *Attack Localizer* whose functions are self-explanatory. For example, these modules can run algorithms for attack detection, identification and localization based on supervised, unsupervised or semi-supervised learning, whose details and experimental performance evaluation can be found in [15]. Based on the attack condition information reported by the ML blocks, a quick attack response action is decided by the *Fast Attack Fixing Processor*. The purpose of this block is to select a quick countermeasure aimed at minimizing service downtime (e.g. by a simple traffic rerouting). The countermeasure implementation is queried to optical controller by means of an appropriate API. The task of a complete attack remedy is rather different, and it is performed by another block: the *Attack Remedy Processor*. This block uses attack classification and localization information to elaborate an ultimate attack fixing strategy. The nature of optical layer attacks implies that the network infrastructure has been fraudulently modified and it is, therefore, unlikely that the attack can be permanently fixed with elementary network functions like traffic protection or rerouting. This is the reason why the Attack Remedy Processor has a suitable API that provides to the SOC systems the information required to organize a human repair intervention on the network infrastructure, such as the physical location of the attack, the attack type, and the type of a physical device that has likely been introduced by a hacker and should be removed. This intervention may consist in immediate actions like switching off optical amplifiers for isolating a link under attack, followed by a physical removal of the attack devices in the field.

As shown in Fig. 2, the OSM can be easily integrated in the general architecture of a Transport Software Defined Network (T-SDN) controller (see for instance [4] where only the Proactive fault detection module was considered) and it exploits some basic functions already present in the architecture: the Northbound and Southbound interfaces and the Telemetry and Telemetry DB. From the functional viewpoint, the OSM and the Proactive fault detection module are designed as independent modules: they both use the information of the Telemetry DB and the services provided by the interfaces, but they work independently and provide to the network management applications different kind of information. This functional independence of the OSM with respect to other network management blocks eases its integration in the T-SDN controller.

D. Optical Security Manager features and requirements

The proposed OSM functions impose different Key Performance Indicators (KPIs) on the various blocks, discussed as follows.

- **Telemetry**: OPM data acquisition time is the most important KPI for telemetry. It also represents the update time of the network attack status reported by ML algorithms.
- **Attack detection and classification**: The detection time and classification time are the basic KPIs together with accuracy of the ML algorithms, in particular the obtained False Positive (FP) and False Negative (FN) rates.
- **Attack reaction**: We can define the KPI of this function as the time required to identify the proper quick reaction strategy (e.g. by a search on a data base of strategies defined "a priori") and to send the corresponding implementation request to the optical controller.
- **Attack localization**: The most important KPI for this function is the attack localization accuracy: for instance, a localization accuracy of a few tens of meters makes the attack device hunting in urban areas quite easy for a field team. Other KPIs are the localization time and the probability of localization error.
- **Permanent attack remediation**: The most important KPI of this function is the time required to provide the fixing information to the SOC systems.

3. EFFICIENT ATTACK COGNITION APPROACHES

Attack cognition is a key capability of the network security management framework. In particular, the attack detection task can be directly related to the classical anomaly detection task in ML. Among the ML techniques, SSL and UL are the most promising ones to implement detection of evolving attacks due to their ability to detect previously unseen attacks. In the following, we introduce the ML techniques for detection of unseen attacks, the main ways to measure accuracy of these techniques, and the dimensionality reduction methods that can be used in combination with the ML techniques.

A. ML Techniques for Detection of Evolving Attacks

One of the key characteristics of SSL and UL techniques is that they do not need labeled data. This characteristic is particularly important in the context of network security, where labeling data requires specialist knowledge for the known attacks, while being impossible for unseen attacks. This means that even if SL models can accurately detect and identify attacks, they can do so only for previously recorded attack methods whose effects have been scrutinized by experts.

ML techniques that do not require labeled data are a compelling solution for coping with the virtually open-ended, quickly-evolving threat space. However, the benefit of not needing a labeled dataset comes at the expense of lower accuracy, inability to provide fine-granular interpretation of e.g. type and intensity of an attack, and typically longer time needed to process the same amount of data in comparison to SL, calling for approaches that improve the efficiency and performance of SSL and UL techniques.

A.1. Semi-Supervised Learning (SSL)

SSL models belong to a category of ML models that do not require a labeled dataset representing all possible conditions, but only a dataset that contains samples considered normal. These models learn the boundaries of the normal working conditions region, which enables detection of data points that fall outside the region. Once the boundaries are learned, the model can perform the inference by only analyzing new samples.

In the context of physical layer optical network security, a drawbacks of SSL models is the need to train a new model or update an existing one upon establishing each new lightpath. This is necessary to accommodate for the large variations in properties of lightpaths that traverse different links, use different transceivers, and are allocated different parts of the optical spectrum. Therefore, the SSL models need to incorporate the normal working conditions of each individual lightpath.

One of the most regarded SSL techniques is the One-Class Support Vector Machine (OCSVM), which uses a kernel function to create a multi-dimensional space. There are three main parameters to be selected when configuring the OCSVM model: *(i)* the *kernel*, which specifies the kernel type used in the algorithm; *(ii)* the γ, which specifies the kernel coefficient for some kernels; and *(iii)* the ν, which specifies an upper bound on the fraction of training errors. During the training phase, the algorithm encloses the normal working condition data as tightly as possible. During inference, if a new data point falls outside of the boundaries of the learned space, it is considered an anomaly (i.e., an attack in the context of this work).

A.2. Unsupervised Learning (UL)

UL models rely on the assumption that anomalies (i.e., attacks) are rare events. Based on this assumption, UL models analyze a significant number of samples at every inference to determine what can be considered as a normal working condition, and what cannot. In the context of this work, this intuition is useful because it bypasses the need to train models. The characterization of normal working conditions is obtained solely by the consideration of a significant number of samples. On the other hand, since a number of samples needs to be analyzed at every inference, UL models tend to be more complex than SSL models, resulting in significantly longer times to make inferences.

One of the most commonly used UL techniques is the Density-Based Spatial Clustering of Applications with Noise (DBSCAN). DBSCAN uses a notion of neighborhood around each sample, and counts how many other samples fall within this neighborhood region. There are two main parameters to be selected when configuring the DBSCAN model: *(i)* the ϵ defines the radius of a neighborhood around each sample; and *(ii)* the $MinPts$ defines how many neighbors a sample should have to be considered a normal sample. Based on these two parameters, samples that do not have enough neighbors are considered anomalies (i.e., attack samples in the context of this work). Another important parameter to consider when using DBSCAN is the size of the sample window, which defines how many previously collected samples are provided to the algorithm to characterize the normal vs. anomalous working conditions. This window should be large enough to include normal variations of parameters over time, without excessively increasing the algorithm complexity which is proportional to the window size.

B. Accuracy Measures for Attack Detection

The accuracy of attack detection techniques can be measured in terms of four basic metrics:

- True negative rate [$T_n \in (0,1)$]: the portion of normal operating condition samples detected as normal samples;
- False positive rate [$F_p \in (0,1)$]: the portion of normal operating condition samples detected as attacks;
- True positive rate [$T_p \in (0,1)$]: the portion of attack samples detected as attacks;
- False negative rate [$F_n \in (0,1)$]: the portion of attack samples detected as normal samples.

The sum of the true negative and the false positive rates must be equal to one, i.e., $T_n + F_p = 1$, as does the sum of the true positive and the false negative rates, i.e., $T_p + F_n = 1$.

Within the scope of our work, it is expected that the number of normal operating condition samples is much greater than the attack ones, configuring a highly imbalanced dataset. In such cases, *precision* and *recall* can be used to summarize the accuracy of the model. Precision [$P \in (0,1)$] defined in Eq. (1) measures the sensitivity of the model under evaluation to false positives. Recall [$R \in (0,1)$] defined in Eq. (2) measures the sensitivity of the model under evaluation to false negatives. Finally, the f1 score [$F1 \in (0,1)$] defined in Eq. (3) computes the harmonic mean of precision and recall, summarizing the accuracy of a model in a single metric. The f1 score is particularly useful because, to achieve a good score, the model must achieve both high precision and high recall.

$$P = \frac{T_p}{T_p + F_p} \tag{1}$$

$$R = \frac{T_p}{T_p + F_n} \tag{2}$$

$$F1 = 2\frac{P \times R}{P + R} \tag{3}$$

C. Dimensionality Reduction Methods

Dimensionality reduction methods have been developed primarily for use in data representation and visualization. These methods assume that most of the useful knowledge of a dataset is usually (or can be) concentrated in only a few of its features or can be summarized with only a few features synthesized from the dataset. This assumption is particularly relevant for optical network monitoring, where DSP-enabled transponders collect a rich set of features, but not all of them are equally indicative for every use case. Dimensionality reduction methods are capable of analyzing datasets with a multitude of dimensions and extracting only a given number of them that contains the most information about the dataset [33]. This allows, for instance, to

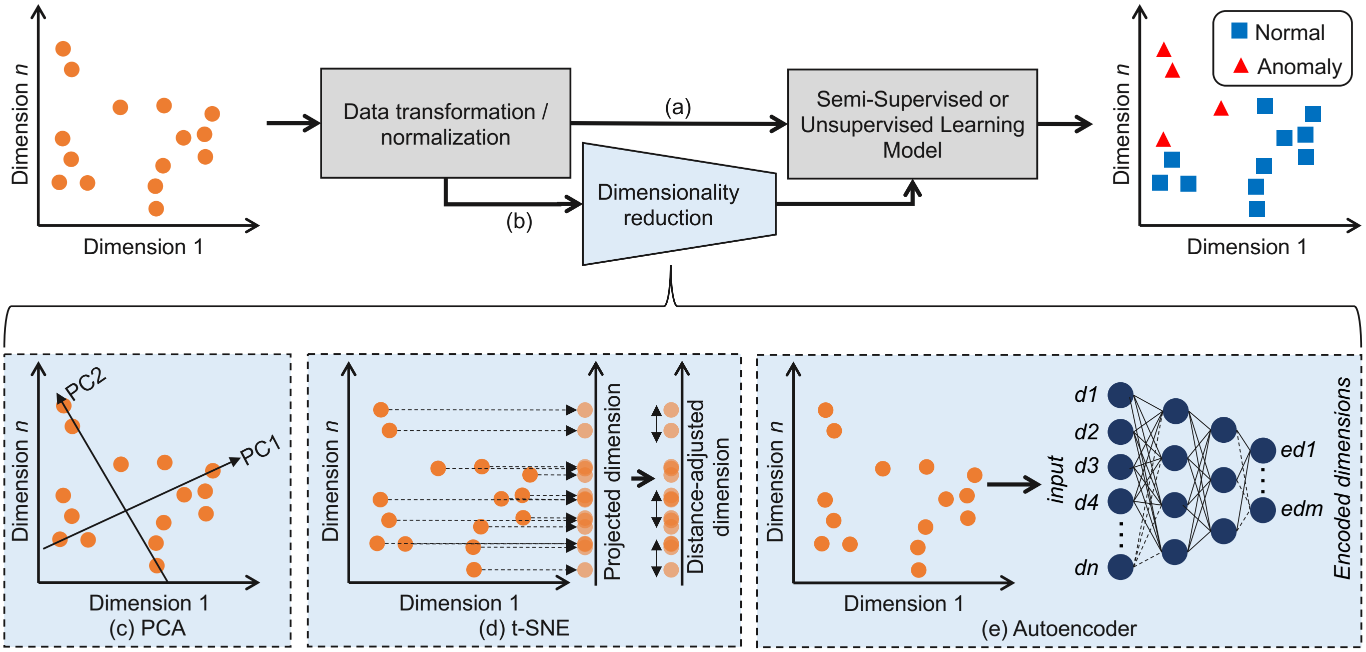


Fig. 3. Workflow of the (a) traditional and (b) dimensionality-reduction-assisted SSL and UL, e.g., (c) PCA, (d) t-SNE and (e) Autoencoder.

visualize a dataset using only two or three dimensions, making it easier for humans to understand the data.

Meanwhile, SSL and UL models may suffer from scalability issues such as run time complexity and sensitivity to high-dimensional data. The run time issue is caused by the fact that, mainly in UL models, the model needs to iterate over a set of samples to evaluate a certain definition of distance between samples, causing its run time to increase exponentially with the number of samples. Sensitivity to high-dimensional data is characterized by cases where increasing the number of features degrades the model accuracy. Models sensitive to dimensionality usually show improved performance when increasing the number of dimensions from one to a few, but degraded performance when the number of dimensions increases further. Conversely, the performance of models that are not sensitive to dimensionality usually improves when the number of dimensions increases from one to a few, reaching a plateau if the number of dimensions is further increased.

Scalability issues of semi-supervised and unsupervised learning models, both in terms of run time complexity and sensitivity to data dimensionality, challenge their application to large, high-dimensional datasets [20, 34, 35]. Such datasets have been successfully analyzed by SL models assisted by DL techniques. However, unlike DL models, SSL and UL models do not have feature extraction capabilities. This means that if there are noisy or irrelevant features in the dataset, these features will degrade the performance of the model. Therefore, dimensionality reduction methods have recently been combined with semi- and unsupervised models as a way to mitigate their sensitivity to high-dimensional data, acting as feature extraction before the execution of the models [19, 35]. It is important to note, however, that potential benefits of applying dimensionality reduction methods are highly dependent on the problem, i.e., on the features and anomalies being detected. Therefore, their application should be evaluated for each problem. In this work, we focus on evaluating their suitability for physical layer attack detection in optical networks.

Fig. 3 shows the workflow of SSL and UL models. In a traditional workflow (Fig. 3(a)) the data is pre-processed before being fed to an SSL or a UL model. The pre-processing applies some variation of data transformation or normalization, which usually consists of fitting the feature values into a scale that facilitates the learning by the models. However, this process does not change the dimensions of the data, i.e., it maintains the same number of features as the original data. Therefore, although data transformation or normalization usually improves the performance (by increasing accuracy and/or facilitating the learning) of the ML models, it does not solve other issues such as sensitivity to high-dimensional data. In a workflow that uses dimensionality reduction (Fig. 3(b)), the dimensionality reduction method computes a few dimensions that better represent the data. Thus, the ML model can concentrate on fewer dimensions when detecting anomalies. The reduced number of dimensions can bring higher accuracy and/or improved scalability to the model [19, 34]. In the following, we describe three of the most used dimensionality reduction methods.

C.1. Principal Component Analysis (PCA)

PCA is a dimensionality reduction method that tries to encode in the extracted features as much variance from the original dataset as possible [36], illustrated in Fig. 3(c). For this purpose, it selects a given number of orthogonal components from the dataset based on their variance. For each component to be extracted from the dataset, PCA finds a linear combination of the original features that results in the highest variance of the projected data points. Once the components are selected, the method can be used to project high-dimensional data onto a lower number of components using a linear combination of the original features. As PCA works with projected-dimension-wise steps (instead of, for instance, sample-wise steps), it works efficiently over large number of samples with large number of dimensions in the dataset.

C.2. t-distributed Stochastic Neighbor Embedding (t-SNE)

t-SNE is a dimensionality reduction method that builds upon the Stochastic Neighbor Embedding with the capability of retaining local and global relations between data points [37], illustrated in Fig. 3(d). t-SNE starts by computing the pair-wise distances among all samples in the original dataset and then randomly projects the data points over a given number of dimensions. t-SNE proceeds with a (predefined or not) number of steps. At each step, the projected data points are moved such that the distance between points which are close in the original dataset is minimized in the projected dimensions. Conversely, the distance between points which are distant in the original dataset is maximized. The direction and amplitude of adjusting the position of each data point (known as gradient) can be computed using different algorithms such as stochastic gradient descent. As t-SNE works with sample-wise steps (as opposed to projected-dimensions-wise, for instance), its complexity (and therefore run time) is expected to be much higher than other dimensionality reduction methods such as PCA and Autoencoders (AEs). There are also a few simplification methods that reduce the complexity of the t-SNE, making it more suitable for very large datasets with many features [38].

C.3. Autoencoder (AE)

Autoencoders are a type of a neural network that can be used for several different purposes, including denoising, one-shot learning and dimensionality reduction [39, 40], illustrated in Fig. 3(e). For the autoencoder, a neural network is used where input and output layers have the same dimension as the dataset. Among the hidden layers, the central layer is the one responsible to contain the encoded representation of the data. The objective of the autoencoder is for the neural network to reproduce at the output the same values it receives at the input. The training is performed in a supervised manner, where the same samples are used both as input as well as the ground truth values at the output.

When an autoencoder is built with the task of performing dimensionality reduction, the feature extraction part is done by a central layer that represents the number of features to be extracted from the dataset. Once the autoencoder is trained, the neural network is split into encoder and decoder parts, and only the encoder part is used to perform the dimensionality reduction. The values output by the encoder can then be directly input to an SSL or a UL model to perform anomaly detection. The AE needs to be trained only once for each lightpath, at the beginning of its operation. In fact, the same trained AE can be used for multiple lightpaths, but this investigation is out of the scope of this work. Since the inference of the AE is quite efficient (and can be further assisted by specific-purpose hardware), its overhead in terms of additional run time is expected to be lower than other methods such as t-SNE.

4. PERFORMANCE ANALYSIS

In this section, we investigate the impact of the three dimensionality reduction methods on the accuracy and scalability of SSL and UL models for physical-layer attack detection in optical networks. The introduction of a dimensionality reduction method into the ML workflow needs to be carefully analyzed by evaluating *(i)* whether the dimensionality reduction improves the accuracy and *(ii)* whether the resulting reduction in the ML model complexity compensates for the extra complexity introduced by the dimensionality reduction method.

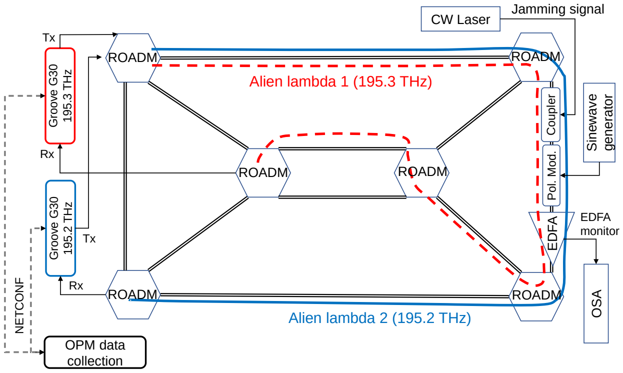


Fig. 4. Network testbed used in security experiments [15].

We carry out this analysis on an experimental optical network security dataset reported in our recent work [15]. The experimental optical network testbed and the dataset characteristics are summarized in subsection A. The impact of the dimensionality reduction methods on the ML model accuracy and run time complexity are analyzed in subsections B and C. The accuracy and run time performance analyses are then combined in subsection D to provide insight into the overall efficiency and tradeoffs involved in designing an ML-based physical layer attack detection mechanism for optical networks.

A. Testbed and Parameter Setup

The dataset used in this paper is collected from an experimental optical network testbed based upon commercial optical transport network technology and is composed by 6 Reconfigurable Optical Add-Drop Multiplexer (ROADM) nodes, 1 Erbium-Doped Fiber Amplifier (EDFA) amplification node and 10 optical fiber links, shown in Fig. 4. The coherent transponders collect a rich OPM dataset reporting the average, minimum and maximum values of the following parameters every minute: chromatic dispersion, differential group delay, Optical Signal-to-Noise Ratio (OSNR), polarization dependent loss, Q factor, block errors, Bit Error Rate (BER) and presence of uncorrected blocks before Forward Error Correction (FEC), BER after FEC, received and transmitted optical power and frequency, and the loss of signal alarm status.

In the testbed, we characterize the normal operating conditions and apply three attack techniques: in-band and out-of-band jamming, and polarization modulation attack. We vary the intensity of the attack for each technique, characterizing a light and a strong attack scenario. The Optical Channels (OChs) under test are two 200 Gbit/s polarization multiplexed 16QAM signals at frequencies of 195.2 and 195.3 THz, respectively, operating error-free with 32 dB $\mathrm{OSNR}_{0.1}$. The properties of attack scenarios, i.e., the power and frequency of the Continuous Wave (CW) jamming signal and the amplitude of the 136 kHz sinewave signal driving the fiber squeezer in the polarization modulation attack, are summarized in Table 1. A detailed description of the experiments can be found in [15]. Each of the 7 security conditions (one normal and 6 attack scenarios) is recorded over a 24-hour period. The resulting dataset contains 1440 samples for each scenario, each sample containing 31 features. Finally, the dataset is cleaned (by removing data points with missing features) and standardized.

We use the implementations available on Scikit-learn [41] for the data processing tasks. For the OCSVM, we investigate

Table 1. Summary of attack scenarios [15].

Attack scenario		Jamming signal power	Jamming signal frequency	Fiber squeezer driver amplitude
Out of band jamming	Light	P_0+3 dB	195.1 THz	-
	Strong	P_0+8.7 dB	195.1 THz	-
In band jamming	Light	P_0-10 dB	f_0	-
	Strong	P_0-7 dB	f_0	-
Polarization modulation	Light	-	-	0.3 V
	Strong	-	-	1.6 V

P_0 and f_0 denote the power level and frequency of the OCh under test.

a number of configurations defined by the kernel, γ and ν parameters.The combinations of the kernels {*rbf*, *linear*, *sigmoid*}, γ values {0.001, 0.1, 0.2, 0.5, 0.7, 1} and ν values {0.01, 0.1, 0.3, 0.5, 0.7, 1} were tested. For the DBSCAN, we investigate a number of configurations defined by the ϵ and $MinPts$ parameters. The combinations of the ϵ values {0.1, 0.5, 1, 2, 3, 4, 5, 10} and $MinPts$ values {3, 5, 8, 10} were tested. Moreover, we assume a window of 100 samples to characterize the normal working conditions of a lightpath in DBSCAN, with a 10:1.5 ratio between normal and attack samples. Naturally, in OCSVM, a lightpath requires the processing of a single monitoring sample.

By applying the dimensionality reduction methods, the dataset is represented by 1 to 7 dimensions. For PCA, the only parameter to be set is the number of components to be extracted. For t-SNE, we additionally set the algorithm to use the exact method to compute the gradients and a maximum of 300 iterations. For AE, after testing several neural network architectures, the architecture with {31, 400, 100, 40, n, 40, 100, 400, 31} neurons (where n represents the number of dimensions to be extracted) was the one that achieved the best performance, i.e., the lowest error. The trainable parameters of the autoencoder are initialized using the uniform initializer [42], and during training the training and validation errors are monitored to ensure that overtraining is not reached. As a baseline for comparison, we tested the performance of the algorithms when supplied with the Full Dataset (FD).

B. Accuracy Analysis

In this section, we focus on assessing the impact of dimensionality reduction methods on the accuracy of the SSL and UL models. Therefore, the OCSVM and DBSCAN algorithms are ran in combination with the dimensionality reduction methods with different parameter settings, as well as the full dataset. Each parameter combination results in a false positive and false negative rate, which can be summarized by the f1 score. Out of these performance results, we highlight the accuracy frontiers, while opaque data points in Figs. 5 and 6 represent configurations which are not part of the accuracy frontier.

Fig. 5 highlights the accuracy frontiers for OCSVM in terms of false positive and false negative rates for the three dimensionality reduction methods and the number of dimensions ranging from 1 to 7. We can observe that all three dimensionality reduction methods degrade the OCSVM accuracy compared to the FD scenario. This demonstrates that OCSVM takes advantage of the extra features included in the full dataset This property is also underlined by the fact that extracting only one dimension substantially degrades accuracy compared to a higher number of dimensions or the FD case.

Across all the dimensions and dimensionality reduction methods, the best f1 score is always achieved using the *rbf* kernel. In the case of PCA (Fig. 5a), the highest accuracy is achieved for 6 dimensions, while further increasing the number of dimensions does not improve the algorithm performance. The highest accuracy is achieved with ν=0.01 and γ=1.0. In the case of t-SNE (Fig. 5b), the best accuracy is achieved already with 3 dimensions, ν=0.1 and γ=0.2, and the variation of performance observed with 2-7 dimensions shows that t-SNE is able to extract useful features from the dataset with fewer dimensions than PCA. Finally, applying the autoencoder obtains a progressive increase in the f1 score as we increase the number of dimensions, with 6 and 7 dimensions presenting similar accuracy. The OCSVM configuration that achieves this accuracy is ν=0.01 and γ=1.0 for both 6 and 7 dimensions.

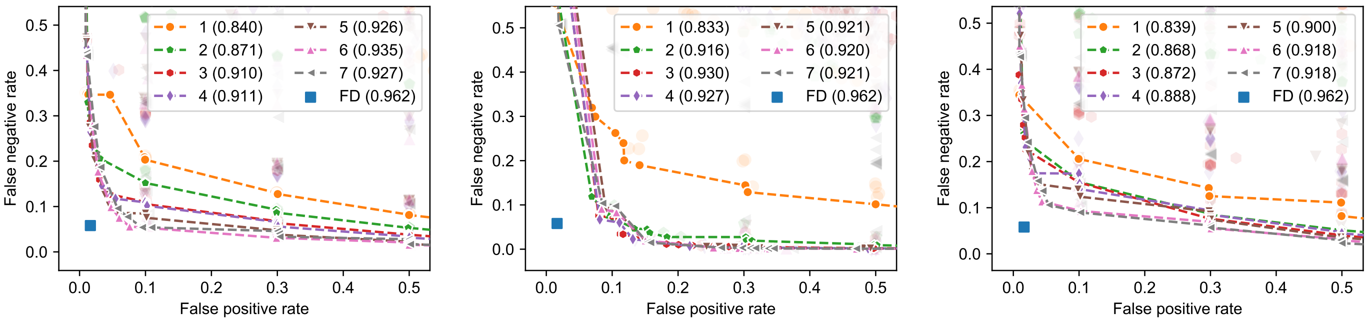
Fig. 6 highlights the accuracy frontiers for DBSCAN. We can observe that, as opposed to OCSVM, dimensionality reduction techniques significantly improve DBSCAN accuracy compared to the FD case. DBSCAN in combination with t-SNE (Fig. 6b) achieves the highest f1 score among the dimensionality reduction methods, and does so with only two dimensions. The reduction in false positive rate reaches as high as 16% and 2.5% in false negative rate with two dimensions when ϵ=1 and $MinPts$=5. Unlike OCSVM which presents a clear gain when the number of dimensions increases, DBSCAN shows a more indefinite trend, requiring careful parameter configuration depending on the particular dimensionality reduction method.

In the case of PCA (Fig. 6a), benefits are obtained only with 4 or more dimensions, with the best accuracy reached when using 6 dimensions, with $MinPts$=10 and ϵ=0.5. The t-SNE, when combined with DBSCAN (Fig. 6b), also brings good accuracy benefits, except for the case with a single dimension. Similar to the behavior observed with OCSVM (Fig. 5b), t-SNE with DBSCAN also shows a smaller accuracy variation for two or more dimensions in comparison with the other dimensionality reduction methods. This demonstrates a trend of t-SNE achieving good accuracy with a lower number of dimensions. The best f1 score is obtained with $MinPts$=10 and ϵ=2.0. The autoencoder also shows benefits in terms of accuracy, but not as substantial as the other two dimensionality reduction methods. Interestingly, using only 3 dimensions already yields the best f1 score (same as for 7 dimensions), with $MinPts$=5 and ϵ=0.1.

C. Run Time Complexity Analysis

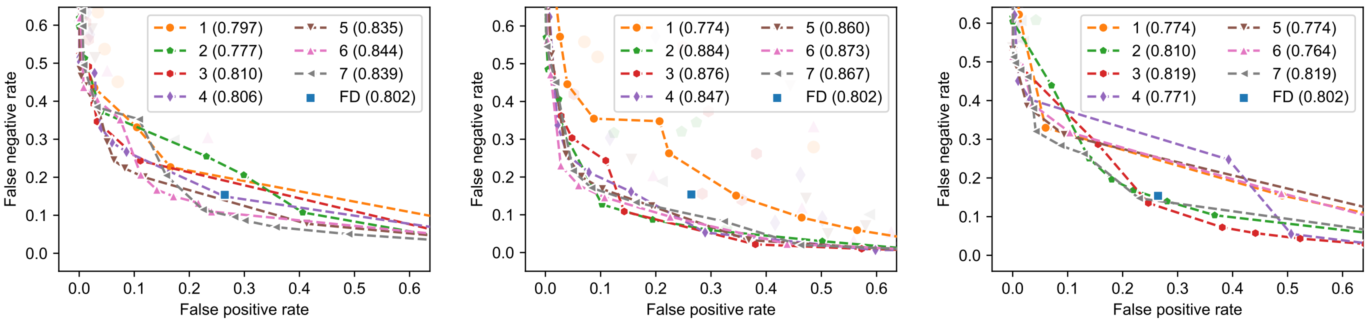
To analyze the run time of the two ML models combined with different dimensionality reduction methods as a function of the number of monitored connections in the network, we have artificially inflated the size of the dataset by generating synthetic samples with each feature following a Gaussian distribution dictated by the average and standard deviation obtained from the original dataset. To conduct a fair comparison, the dataset presented to DBSCAN contains 100 samples per lightpath (necessary to characterize the normal working conditions), while the one presented to OCSVM contains one sample per lightpath. The synthetically generated dataset is then processed by the dimensionality reduction methods (when applicable) and by the ML models, while the run times are recorded.

Fig. 7 shows the amount of time taken by the entire attack detection process, including dimensionality reduction and the



(a) PCA (b) t-SNE (c) Autoencoder

Fig. 5. Accuracy of the SSL (i.e. OCSVM) model for the different dimensionality reduction methods and different number of dimensions. The number in parentheses represents the highest f1 score obtained for the respective number of dimensions and the full dataset. Opaque data points represent configurations which are not part of the accuracy frontier.



(a) PCA (b) t-SNE (c) Autoencoder

Fig. 6. Accuracy of the UL (i.e. DBSCAN) model for the different dimensionality reduction methods and different number of dimensions. The number in parentheses represents the highest f1 score obtained for the respective number of dimensions and the full dataset. Opaque data points represent configurations which are not part of the accuracy frontier.

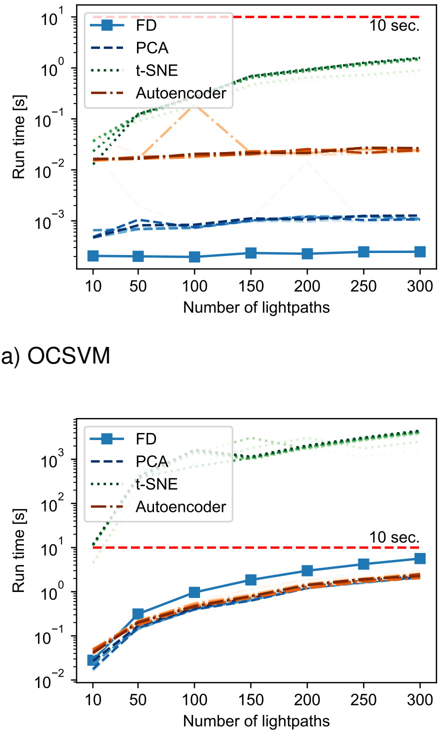
anomaly detection algorithm, for a varying number of monitored lightpaths. The times were obtained by running the algorithms on an Intel Core i9 9900X CPU clocked at 3.5 GHz with 64 GB of RAM. For both OCSVM and DBSCAN, a clear trend in complexity can be observed. PCA is the method that incurs the least extra run time out of the methods, followed by the autoencoder. The t-SNE method shows a much higher overhead, imposing up to three orders of magnitude higher run time than other dimensionality reduction methods.

Fig. 7a shows that when OCSVM is used, there is no run time benefit in using a dimensionality reduction method. In fact, when these results are combined with the ones discussed in the previous section, we can state that there are no benefits in using dimensionality reduction methods in combination with OCSVM. This is substantiated by the fact that the OCSVM can process FD data associated to 300 lightpaths in 0.2 milliseconds. Fig. 7b indicates an opposite trend for DBSCAN. While t-SNE imposes a significant runtime overhead, PCA and autoencoder can reduce run time by a significant margin, i.e., by almost one order of magnitude compared to FD. This reduction represents the ability to process a few hundreds more lightpaths in a real-world deployment if we consider a 10 seconds monitoring loop. Moreover, combined with an improved accuracy brought by the dimensionality reduction methods, DBSCAN can greatly benefit from the introduction of these methods.

D. Accuracy vs. Run Time

The accuracy and run time performance of the OCSVM and DBSCAN models with and without the dimensionality reduction methods are summarized in Fig. 8. We can see that the OCSVM performance is concentrated in the bottom right part of the plot, while DBSCAN is concentrated in the top left part. This clearly shows that the SSL model, i.e. OCSVM, is a more accurate and a more time efficient option for physical layer attack detection in optical networks. Conversely, UL, i.e. DBSCAN, achieves lower accuracy and takes longer to run.

At first, these properties might seem to indicate that SSL is the only model of choice for detection of novel types of physical layer attacks in optical networks while UL has no advantages. However, inaccuracies can be mitigated by the use of several methods, such as a window-based approach [15]. Moreover, in a real-world deployment, SSL models may introduce extra complexity beyond their use to detect attacks, such as the need of training the model for every new lightpath. Therefore, it might make sense for network operators to sacrifice accuracy and run time complexity for easier use during operation, depending on the network characteristics and specific use cases.



(a) OCSVM

(b) DBSCAN

Fig. 7. Run time (in seconds) necessary for processing a number of lightpaths. FD accounts for the model run time using all the features. PCA, t-SNE and AE account for the run time of the dimensionality reduction and the model run time. Different shades of colors denote different number of dimensions.

5. OPEN CHALLENGES AND FUTURE WORK

Achieving truly autonomous, multi-layer network security management requires tackling several remaining open challenges. A set of these challenges is related to the human-ML interaction. While the refinements in ML performance may reduce the level of human interventions, their elimination is highly unlikely. Automation of suitable data processing tasks and attack remediation strategies, development of visualization platforms, and improving explainability of ML outputs, will be key to unburdening the security experts and allowing them to create advanced solutions for unprecedented circumstances.

Multi-domain security management will require exchange of relevant data and knowledge on security incidents across different domains without violating confidential or proprietary information. Development of privacy-preserving federated learning models may provide a useful mechanism for collaborative training without sharing possibly sensitive data. However, such models are prone to adversarial attacks such as data poisoning which target the ML algorithms themselves. If an attacker deceives an ML algorithm into falsely detecting attacks, the unnecessarily triggered response can inflict substantial damage to network operators in terms of e.g. enlarged operating expenditures. Use of smart contracts or distributed ledger technologies such as blockchain may help counteract this by enforcing ML models' privacy and trustworthiness [43].

Tight integration of optical-layer security with existing SOCs frameworks will require approaches that are capable of dealing with uncertainties stemming from both the ML models and the observed environment. The time scale of tasks and actions impacted by these uncertainties needs to be taken into account when deciding on the most appropriate form of modeling. For example, less time-critical tasks with a lower level of uncertainty can benefit from adaptive automation of converting learning models to algorithms, processes and workflows, while more time-critical ones with greater sensitivity require acceleration techniques and continuous incremental learning [44].

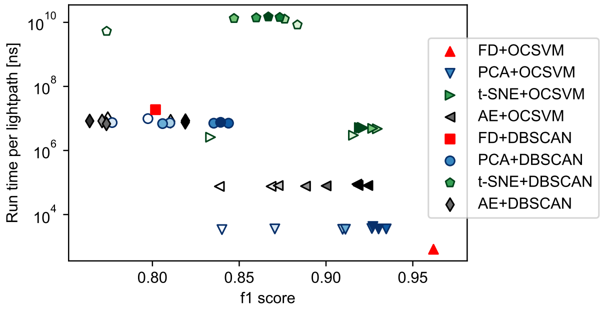


Fig. 8. Run time per lightpath (in ns) vs. accuracy (best f1 score) for the Full Dataset (FD), PCA, t-SNE and AE dimensionality reduction methods combined with OCSVM and DBSCAN models.

6. CONCLUSIONS

This paper focuses on challenges related to the incorporation of optical network security assurance into carrier-grade network management processes. To tackle practical system integration issues, a new functional block called Security Operation Center (SOC) was proposed, along with defining its architecture, functionalities, and KPI requirements. To cope with the evolving intelligent adversary landscape, unsupervised and semi-supervised ML techniques were applied for detection of previously unseen attacks. Their sensitivity to the OPM dataset dimensionality was assessed and they were combined with dimensionality reduction methods to analyze the trade-offs between the obtained accuracy and run time complexity, aiding network operators to make informed choices on optical network security management. The results showed that SSL might not benefit from dimensionality reduction methods, while for UL benefits in terms of both increased accuracy as well as reduced run time were observed.

FUNDING

Vetenskapsrådet (2019-05008).

ACKNOWLEDGMENTS

The authors gratefully acknowledge Roberto Morro for fruitful discussions.

REFERENCES

1. T. Uematsu, H. Hirota, T. Kawano, T. Kiyokura, and T. Manabe, "Design of a temporary optical coupler using fiber bending for traffic monitoring," IEEE Photonics J. **9**, 1–13 (2017).

2. T. Szyrkowiec, M. Santuari, M. Chamania, D. Siracusa, A. Autenrieth, V. Lopez, J. Cho, and W. Kellerer, "Automatic intent-based secure service creation through a multilayer SDN network orchestration," IEEE/OSA J. Opt. Commun. Netw. **10**, 289–297 (2018). DOI: 10.1364/JOCN.10.000289.
3. S. Zeadally, E. Adi, Z. Baig, and I. A. Khan, "Harnessing artificial intelligence capabilities to improve cybersecurity," IEEE Access **8**, 23817–23837 (2020). DOI: 10.1109/ACCESS.2020.2968045.
4. D. Rafique, T. Szyrkowiec, H. Grießer, A. Autenrieth, and J.-P. Elbers, "Cognitive assurance architecture for optical network fault management," IEEE/OSA J. Light. Technol. **36**, 1443–1450 (2018). DOI: 10.1109/JLT.2017.2781540.
5. T. Tanaka, A. Hirano, S. Kobayashi, T. Oda, S. Kuwabara, A. Lord, P. Gunning, O. Gonzáles de Dios, V. Lopez, A. M. Lopez de Lerma, and A. Manzalini, "Autonomous network diagnosis from the carrier perspective [invited]," IEEE/OSA J. Opt. Commun. Netw. **12**, A9–A17 (2020). DOI: 10.1364/JOCN.12.0000A9.
6. L. Velasco, A. C. Piat, O. González, A. Lord, A. Napoli, P. Layec, D. Rafique, A. D'Errico, D. King, M. Ruiz, F. Cugini, and R. Casellas, "Monitoring and data analytics for optical networking: Benefits, architectures, and use cases," IEEE Netw. pp. 1–9 (2019). DOI: 10.1109/MNET.2019.1800341.
7. I. Sartzetakis, K. K. Christodoulopoulos, and E. M. Varvarigos, "Accurate quality of transmission estimation with machine learning," IEEE/OSA J. Opt. Commun. Netw. **11**, 140–150 (2019). DOI: 10.1364/JOCN.11.000140.
8. M. Lonardi, J. Pesic, P. Jennevé, P. Ramantanis, N. Rossi, A. Ghazisaeidi, and S. Bigo, "Optical nonlinearity monitoring and launch power optimization by artificial neural networks," J. Light. Technol. **38**, 2637–2645 (2020). DOI: 10.1109/JLT.2020.2985779.
9. F. Musumeci, C. Rottondi, G. Corani, S. Shahkarami, F. Cugini, and M. Tornatore, "A tutorial on machine learning for failure management in optical networks," IEEE/OSA J. Light. Techn. **37**, 4125–4139 (2019). DOI: 10.1109/JLT.2019.2922586.
10. B. Shariati, M. Ruiz, J. Comellas, and L. Velasco, "Learning from the optical spectrum: Failure detection and identification," IEEE/OSA J. Light. Techn. **37**, 433–440 (2019). DOI: 10.1109/JLT.2018.2859199.
11. X. Chen, B. Li, R. Proietti, Z. Zhu, and S. J. B. Yoo, "Self-taught anomaly detection with hybrid unsupervised/supervised machine learning in optical networks," IEEE/OSA J. Light. Techn. **37**, 1742–1749 (2019). DOI: 10.1109/JLT.2019.2902487.
12. Y. Li, N. Hua, Y. Yu, Q. Luo, and X. Zheng, "Light source and trail recognition via optical spectrum feature analysis for optical network security," IEEE Commun. Lett. **22**, 982–985 (2018).
13. M. Bensalem, S. K. Singh, and A. Jukan, "On detecting and preventing jamming attacks with machine learning in optical networks," in *2019 IEEE Global Communications Conference (GLOBECOM),* (2019), pp. 1–6. DOI: 10.1109/GLOBECOM38437.2019.9013238.
14. C. Natalino, M. Schiano, A. Di Giglio, L. Wosinska, and M. Furdek, "Experimental study of machine-learning-based detection and identification of physical-layer attacks in optical networks," IEEE/OSA J. Light. Technol. **37**, 4173–4182 (2019). DOI: 10.1109/JLT.2019.2923558.
15. M. Furdek, C. Natalino, F. Lipp, D. Hock, A. D. Giglio, and M. Schiano, "Machine learning for optical network security monitoring: A practical perspective," J. Light. Technol. **38**, 2860–2871 (2020). DOI: 10.1109/JLT.2020.2987032.
16. G. Liu, K. Zhang, X. Chen, H. Lu, J. Guo, J. Yin, R. Proietti, Z. Zhu, and S. J. B. Yoo, "Hierarchical learning for cognitive end-to-end service provisioning in multi-domain autonomous optical networks," J. Light. Technol. **37**, 218–225 (2019). DOI: 10.1109/JLT.2018.2883898.
17. K. Bresniker, A. Gavrilovska, J. Holt, D. Milojicic, and T. Tran, "Grand challenge: Applying artificial intelligence and machine learning to cybersecurity," Computer. **52**, 45–52 (2019). DOI: 10.1109/MC.2019.2942584.
18. M. Filer, J. Gaudette, M. Ghobadi, R. Mahajan, T. Issenhuth, B. Klinkers, and J. Cox, "Elastic optical networking in the microsoft cloud [invited]," IEEE/OSA J. Opt. Commun. Netw. **8**, A45–A54 (2016). DOI: 10.1364/JOCN.8.000A45.
19. E. Schubert, J. Sander, M. Ester, H. P. Kriegel, and X. Xu, "DBSCAN revisited, revisited: Why and how you should (still) use DBSCAN," ACM Trans. Database Syst. **42** (2017). DOI: 10.1145/3068335.
20. Y. Chen, S. Tang, N. Bouguila, C. Wang, J. Du, and H. Li, "A fast clustering algorithm based on pruning unnecessary distance computations in dbscan for high-dimensional data," Pattern Recognit. **83**, 375 – 387 (2018). DOI: 10.1016/j.patcog.2018.05.030.
21. M. Furdek and C. Natalino, "Machine learning for optical network security management," in *Optical Fiber Communication Conference (OFC) 2020,* (Optical Society of America, 2020), p. M4E.4.
22. C. Natalino, A. Yayimli, L. Wosinska, and M. Furdek, "Infrastructure upgrade framework for content delivery networks robust to targeted attacks," Opt. Switch. Netw. **31**, 202 – 210 (2019). DOI: 10.1016/j.osn.2018.10.006.
23. N. Skorin-Kapov, J. Chen, and L. Wosinska, "A new approach to optical networks security: Attack-aware routing and wavelength assignment," IEEE Trans. Netw. **18**, 750–760 (2010). DOI: 10.1109/TNET.2009.2031555.
24. K. Manousakis, P. Kollios, and G. Ellinas, "Multi-period attack-aware optical network planning under demand uncertainty," in *Optical Fiber and Wireless Communications,* R. Roka, ed. (2017). DOI: 10.5772/intechopen.68491.
25. J. Zhu, B. Zhao, and Z. Zhu, "Leveraging game theory to achieve efficient attack-aware service provisioning in EONs," IEEE/OSA J. Light. Techn. **35**, 1785–1796 (2017). DOI: 10.1109/JLT.2017.2656892.
26. Tao Wu and A. K. Somani, "Cross-talk attack monitoring and localization in all-optical networks," IEEE/ACM Trans. Netw. **13**, 1390–1401 (2005). DOI: 10.1109/TNET.2005.860103.
27. M. Furdek, V. W. S. Chan, C. Natalino, and L. Wosinska, "Network-wide localization of optical-layer attacks," in *Proc. of ONDM,* (Athens, Greece, 2019), pp. 310–322. DOI: 10.1007/978-3-030-38085-4_27.
28. M. Furdek, N. Skorin-Kapov, and L. Wosinska, "Attack-aware dedicated path protection in optical networks," IEEE/OSA J. Light. Techn. **34**, 1050–1061 (2016). DOI: 10.1109/JLT.2015.2509161.
29. Y. Li, N. Hua, Y. Song, S. Li, and X. Zheng, "Fast lightpath hopping enabled by time synchronization for optical network security," IEEE Commun. Lett. **20**, 101–104 (2016). DOI: 10.1109/LCOMM.2015.2497703.
30. N. Sambo, K. Christodoulopoulos, N. Argyris, P. Giardina, C. Delezoide, D. Roccato, A. Percelsi, R. Morro, A. Sgambelluri, A. Kretsis, G. Kanakis, G. Bernini, E. Varvarigos, and P. Castoldi, "Field trial: Demonstrating automatic reconfiguration of optical networks based on finite state machine," J. Light. Technol. **37**, 4090–4097 (2019). DOI: 10.1109/JLT.2019.2922841.
31. E. Riccardi, P. Gunning, O. González de Dios, M. Quagliotti, V. López, and A. Lord, "An operator view on the introduction of white boxes into optical networks," J. Light. Technol. **36**, 3062–3072 (2018). DOI: 10.1109/JLT.2018.2815266.
32. OpenDaylight, "Transport PCE," https://docs.opendaylight.org/en/stable-fluorine/release-notes/projects/transportpce.html.
33. G. E. Hinton and R. R. Salakhutdinov, "Reducing the dimensionality of data with neural networks," Science. **313**, 504–507 (2006).
34. W. Wang, Y. Huang, Y. Wang, and L. Wang, "Generalized autoencoder: A neural network framework for dimensionality reduction," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops,* (2014), pp. 490–497.
35. S. A. Shah and V. Koltun, "Robust continuous clustering," Proc. Natl. Acad. Sci. **114**, 9814–9819 (2017). DOI: 10.1073/pnas.1700770114.
36. R. Vidal, Yi Ma, and S. Sastry, "Generalized principal component analysis (gpca)," IEEE Transactions on Pattern Analysis Mach. Intell. **27**, 1945–1959 (2005). DOI: 10.1109/TPAMI.2005.244.
37. L. v. d. Maaten and G. Hinton, "Visualizing data using t-SNE," J. machine learning research **9**, 2579–2605 (2008).
38. L. Van Der Maaten, "Accelerating t-SNE using tree-based algorithms," The J. Mach. Learn. Res. **15**, 3221–3245 (2014).
39. G. E. Hinton, S. Osindero, and Y.-W. Teh, "A fast learning algorithm for deep belief nets," Neural computation **18**, 1527–1554 (2006). DOI: 10.1162/neco.2006.18.7.1527.
40. Y. Wang, H. Yao, and S. Zhao, "Auto-encoder based dimen-

sionality reduction," Neurocomputing **184**, 232–242 (2016). DOI: 10.1016/j.neucom.2015.08.104.

41. F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, "Scikit-learn: Machine learning in Python," J. Mach. Learn. Res. **12**, 2825–2830 (2011).
42. X. Glorot and Y. Bengio, "Understanding the difficulty of training deep feedforward neural networks," (JMLR Workshop and Conference Proceedings, Chia Laguna Resort, Sardinia, Italy, 2010), pp. 249–256.
43. P. C. M. Arachchige, P. Bertok, I. Khalil, D. Liu, S. Camtepe, and M. Atiquzzaman, "A trustworthy privacy preserving framework for machine learning in industrial iot systems," IEEE Transactions on Ind. Informatics **16**, 6092–6102 (2020). DOI: 10.1109/TII.2020.2974555.
44. G. Cirincione and D. Verma, "Federated machine learning for multidomain operations at the tactical edge," in *Artificial Intelligence and Machine Learning for Multi-Domain Operations Applications,* vol. 11006 T. Pham, ed. (SPIE, 2019), pp. 29 – 48. DOI: 10.1117/12.2526661.