CHALMERS
UNIVERSITY OF TECHNOLOGY

Experimental Study of Machine-Learning-Based Detection and Identification of Physical-Layer Attacks in Optical Networks

Citation for the original published paper (version of record):
Natalino Da Silva, C., Schiano, M., Di Giglio, A. et al (2019). Experimental Study of Machine-Learning-Based Detection and Identification of Physical-Layer Attacks in Optical Networks. Journal of Lightwave Technology, 37(16): 4173-4182.
http://dx.doi.org/10.1109/JLT.2019.2923558

(article starts on next page)

Experimental Study of Machine-Learning-Based Detection and Identification of Physical-Layer Attacks in Optical Networks

Carlos Natalino, *Member, IEEE*, Marco Schiano, *Senior Member, IEEE*, Andrea Di Giglio, Lena Wosinska, *Senior Member, IEEE*, and Marija Furdek, *Senior Member, IEEE, OSA*

***Abstract*— Optical networks are critical infrastructure supporting vital services and are vulnerable to different types of malicious attacks targeting service disruption at the optical layer. Due to the various attack techniques causing diverse physical-layer effects, as well as the limitations and sparse placement of optical performance monitoring devices, such attacks are difficult to detect, and their signatures are unknown.**

This paper presents an experimental investigation of a Machine Learning (ML) framework for detection and identification of physical-layer attacks, based on experimental attack traces from an operator field-deployed testbed with coherent receivers. We perform in-band and out-of-band jamming signal insertion attacks, as well as polarization scrambling attacks, each with varying intensities. We then evaluate 8 different ML classifiers in terms of their accuracy, and scalability in processing experimental data. The optical parameters critical for accurate attack identification are identified and the generalization of the models is validated. Results indicate that Artificial Neural Networks (ANNs) achieve 99.9% accuracy in attack type and intensity classification, and are capable of processing 1 million samples in less than 10 seconds.

***Index Terms*— optical network security, monitoring, machine learning, attack detection.**

I. Introduction

As the only future-proof technology capable of sustaining the pressing growth rates of network traffic, optical networks represent critical communication infrastructure supporting a wide range of vital societal services. As such, they make an attractive target of attacks aimed at exploiting physical-layer vulnerabilities to disrupt services by, e.g., inserting harmful signals or disabling critical components [1] via direct access to the fiber plant deployed in unprotected environments such as ducts and manholes. Reported occurrences of severe optical network security breaches can be found in [2], [3]. The damage caused by service disruption attacks can escalate from immediate deterioration of the transmission quality of Wavelength Division Multiplexing (WDM) data channels to performance degradation of the carried upper-layer services in a cascading fashion. Optical-layer security issues become even more significant for the proliferating advanced physical-layer paradigms such as Quantum Key Distribution (QKD) and Space Division Multiplexing (SDM) with high-core-count multi-core fibers, which are highly sensitive to physical-layer disturbances aggravated by attacks. Combined with the ultra-high data rates carried in today's networks and stringent performance requirements of next-generation services, all of the above calls for a high degree of operators' preparedness to optical-layer security breaches.

The most disruptive attack methods reported in the literature include in-band and out-of-band jamming, where a harmful signal is inserted in the fiber. Aside from directly accessing the patch-panel and tampering with the fiber plant, this type of attack can be performed by bending the field-deployed fiber and creating a temporary coupler, according to the method described in [4] and depicted in Fig. 1. An in-band jamming signal overlaps with the useful optical channel and adds unfilterable noise. An out-of-band jamming signal inserted

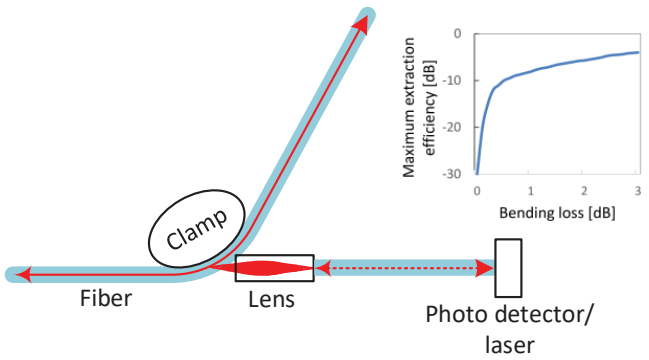


Fig 1. Creating a temporary coupler to extract or insert a signal [4].

Manuscript received December 13, 2018; revised April 28, 2019. (*Corresponding author: Marija Furdek.*) This article is based upon work from COST Action 15127 RECODIS and Celtic-Plus project SENDATE-EXTEND. We gratefully acknowledge Infinera for providing the Groove G30 transponder.

C. Natalino, L. Wosinska and M. Furdek are with the Optical Networks Unit, Dept. of Electrical Engineering, Chalmers University of Technology, Gothenburg, Sweden (e-mail: {carlos.natalino, wosinska, furdek}@chalmers.se).

Marco Schiano and Andrea Di Giglio are with Telecom Italia, Turin, Italy (e-mail: {marco.schiano, andrea.digiglio}@telecomitalia.com).

outside of the used spectrum can degrade the quality of co-propagating channels via gain reduction in optical amplifiers and intensified physical-layer impairments in optical fibers and switches. The intensity of the effects of the inserted signals and the resulting damage depends on their frequency and power levels, the components deployed in the network, as well as on the routes, modulation formats and spectrum assigned to the legitimate connections. In addition to the deliberate insertion of harmful signals into the fiber, service disruption attacks can be implemented by techniques that do not require fiber intrusion. An example is the polarization modulation attack, demonstrated in this paper for the first time, which applies a fast-varying lateral pressure on the fiber under attack and thus inflicts fast polarization scrambling on the propagating signals. When this polarization scrambling is fast enough (typically above a few million radians per second), the polarization recovery algorithms of coherent receivers are no longer able to demultiplex the two orthogonally polarized signals, which results in errors. A polarization modulation attack can be implemented by applying to the bare fiber a relatively simple clip-on device with piezoelectric actuators similar to the one used in our experiment and described in Section IV. Unlike the bent-fiber couplers, this device does not introduce additional loss and is very difficult to locate.

The diversity of the mechanisms exploited by different attack methods, and their disparate effects on legitimate optical channels make detection and identification of physical-layer attacks extremely challenging. Models of physical-layer attacks are scarce and simplistic, unable to capture the complex effects of a range of attacks. As attacks may cause optical parameters to deviate from regular operating conditions, simulating attack behavior using the well-known theoretical models of physical-layer impairments may fall short of capturing their complex effects, which motivates the need for an experimental testbed investigation. Another challenge in security diagnostics stems from the prohibitively high cost of Optical Performance Monitoring (OPM) equipment, which leads to sparse deployment of diverse equipment that does not ubiquitously provide a consistent set of monitoring parameters. The latter issue is somewhat alleviated by OPM-enabled coherent transceivers which allow operators to collect an extensive set of real-time measurements and monitoring data at the destination nodes that can then serve to interpret the security status of each signal. However, modeling the effects of physical-layer attacks and identifying attack signatures, which is a prerequisite for their quick and accurate detection (and subsequently, effective network recovery), still pose a major challenge.

The complex explicit characterization of effects caused by physical-layer attacks makes optical network security assessment ideal for the application of ML methods. ML is well-suited for processing huge amounts of data and identifying intricate patterns among them without the need for explicit specification of models or parameter thresholds, and it has shown great potential for enhancing optical network performance [5], [6]. Recent research has brought forth several applications of ML to optical networking issues such as soft-fault identification [7], [8] and predictive maintenance [9]. However, the application of ML to optical network security diagnostics is still in its infancy. In this paper, extending upon our preliminary study in [10], we experimentally investigate an ML-based framework for detection and identification of attacks targeting disruption of the optical layer. To the best of our knowledge, this is the first experimental demonstration of several physical-layer attack techniques in an operator testbed, and the first application of ML tools to distinguish among diverse attack techniques of different intensities, making an important step towards improving physical-layer security of optical networks.

Our proposed framework is based on extensive measurements gathered from a field-deployed experimental network where OPM data is collected by a coherent transceiver under normal operating conditions and in the presence of *(i)* an in-band (IB) jamming signal, *(ii)* an out-of-band (OOB) jamming signal inserted in the network, and *(iii)* a polarization modulation attack. For each of these attacks, two different intensity levels are considered, i.e., light and strong. The problem is modeled as a classification problem aimed at detecting the type of attack and identifying its intensity. We evaluate the performance of eight different ML algorithms for the attack detection and identification framework, all of them trained with the collected OPM data. Besides the training, validation and testing, we also perform cross-validation in order to evaluate the reliability of the classifiers. We assess the accuracy of the ML classifiers with respect to the OPM parameters that contribute the most to the classification, with the goal of finding the smaller set of parameters that can provide the maximum accuracy. To understand the impact of different OPM parameters (or features) to the classification accuracy, we analyze the classifiers' accuracy for varied subsets of features, and extract the average, upper, and lower bounds on the accuracy for different missing features. Moreover, we analyze the ML classifiers' scalability for a varied number of samples to verify the attack detection time. Among the proposed approaches, the one based on ANNs detects attacks with 99.9% average accuracy over all attack scenarios, and is capable of processing up to 1 million samples in less than 10 seconds using ordinary off-the-shelf hardware.

The remainder of the paper is organized as follows. Section II reviews the related work on optical-layer security and application of ML to optical network management. Section III presents our proposed framework for attack detection and identification. Section IV describes the experimental testbed and measurement procedures used to emulate attacks in real-life scenarios. Section V analyzes the performance of the proposed ML-based attack identification framework, while Section VI concludes the paper.

II. Literature Overview

Systematic overviews of physical-layer vulnerabilities and attack techniques that exploit them in order to disrupt services

or perform unauthorized access to data carried by optical networks can be found in [1], [11]. Management of optical-layer security has been in focus of substantial research efforts worldwide. It can broadly be classified into [12]: *(i)* security assurance through modeling of attack consequences and attack surface minimization [13]–[16]; *(ii)* security assessment through monitoring and detection of attacks, which is the primary goal of this work; and *(iii)* attack recovery through re-configuration of affected connections, attack source neutralization and network adaptation [17], [18].

Security assessment in optical networks comprises the diagnostics of the security status of individual connections and network-wide localization of the attack source. Network-wide attack localization has been addressed in the context of high-power jamming with the main objective of identifying the lightpath that carries the harmful signal [19], [20]. These approaches track power surges generated by an attack along each connection and create alarm trees associated to different origins of attacks. They rely on the availability of accurate real-time status of all connections at the input and the output ports of all nodes in the network, which is a costly, unrealistic and unscalable assumption. The security cognition potential of the network may be enhanced by strategic placement of optical performance monitors [21], but it requires extra investment from the operator nonetheless. Moreover, the applicability of the above approaches is limited to attacks that cause detectable power surges, which is not the case for all attack scenarios considered in this paper. An alternative approach for network-wide localization of harmful signals which does not require specialized monitoring devices is based on tracking the health of the channels at their receivers, and correlating their statuses to deduce the insertion point of a harmful signal according to the subset of affected connections [22]. However, the technique relies on accurate security status assessment of individual connections, which is not attainable by existing approaches. To the best of our knowledge, no accurate theoretical nor experimental data-driven models of physical-layer attacks exist which could be applied for real-time security diagnostics of distinct optical channels. To close this gap, we perform experimental analysis of realistic attack scenarios on an operator's testbed, and leverage machine learning approaches to analyze the experimentally obtained data in order to detect and identify attacks affecting individual optical channels.

ML algorithms have been widely used to support monitoring and management of optical networks [5], [6]. A recurrent application of ML in optical networks is prediction/estimation of Quality of Transmission (QoT) [23] – [25] of unestablished lightpaths. The ML estimation allows operators to avoid running computationally-intensive algorithms, thus reducing complexity of operating the optical network. ML models have also been used to estimate QoT of quantum channels [26], supporting the coexistence of QKD and WDM channels over the same optical fiber. Another application of ML is the detection, identification and localization of soft-failures [7], [8], [27], [28]. ML has been shown to achieve very high accuracy in detection and identification of failures and anomalies, contributing to the reliable operation of the optical network [9].

An ML technique, i.e., support vector machines, has recently been applied to optical network security in [29], where the authors experimentally investigated cooperative detection of unauthorized signals and their paths in the network by analyzing the optical spectrum features. Besides focusing only on unauthorized signals, that approach requires specialized OSA devices, which are costly and provide limited OPM information. In [30], different ML techniques are applied to detect in- and out-of-band jamming attacks using a dataset obtained via simulations only, and derive attack detection probability as an input to, preemptive resource reallocation for reducing the damage from attacks. Contrary to [29], we propose an approach that relies only on the data collected at the coherent off-the-shelf receivers, where each connection needs to be detected anyway, without requiring any specialized monitoring equipment. This enables cost-efficient detection of security threats reliant on a standardized OPM set, and applicable to an array of attack methods which exploit different mechanisms and affect diverse OPM parameters. Moreover, unlike [30], we apply ML techniques to experimental data obtained from an operator metropolitan testbed.

In this paper, we extend our preliminary study from [10], where we used a real-world testbed to perform out-of-band jamming attacks with different intensities, and applied two different ML classifiers for attack detection and identification. The extension includes an enhancement of the testbed where we now perform three different types of attacks (one of them reported for the first time) of varying intensities. The dataset is collected over a longer period, making it richer and more suitable. The range of ML considered classifiers is broadened, the reliability of the generalized models is evaluated via *k*-fold cross-validation, the OPM parameters critical for accurate diagnostics are identified, and ML performance over datasets with missing features is evaluated. Finally, a scalability assessment demonstrates that the proposed framework can run over a very large network, even considering that monitoring samples are collected every minute.

III. Machine-Learning-Based Attack Detection and Identification

Monitoring capabilities are a key enabler of dynamic and autonomous operation of optical networks [31]. As the amount of information collected from the network increases, more advanced algorithms are required to automate the analysis and self-configuration of these networks. Machine learning arises as one of the most promising ways of handling these data [6]. These technologies can provide comprehensive sensing of the network status, giving cognitive algorithms the ability to self-optimize the network parameters [32]. In this section, we describe the Attack Detection and Identification (ADI) framework proposed in this work and briefly introduce the ML algorithms considered for the framework.

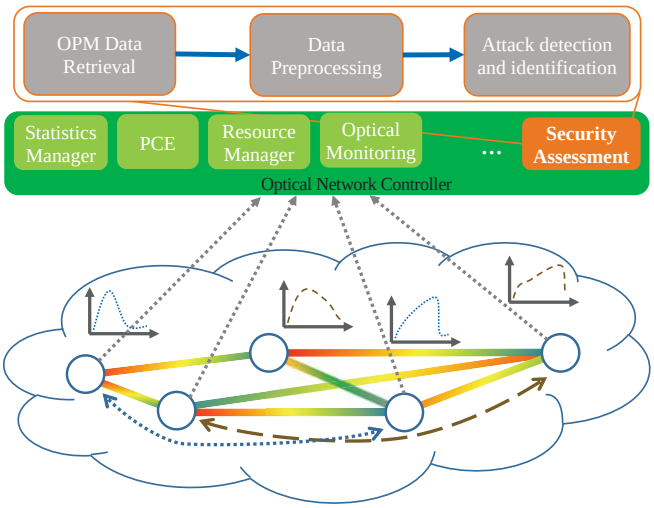


Fig 2. Attack Detection and Identification (ADI) framework.

A. Attack Detection and Identification (ADI) Framework

The ADI framework is depicted in Fig. 2. The framework collects the OPM data from each lightpath established in the network. In this way, it can not only analyze each individual lightpath, but also perform combined analysis considering co-propagating lightpaths.

The framework consists of three steps to be executed each time new monitoring data are available. The first step consists of retrieving OPM data. In the context of ML, each data point in time collected from the lightpaths is referred to as a sample, while each collected OPM parameter is referred to as a feature. In the context of Software-Defined Optical Networking (SDON), the OPM data is retrieved by the optical monitoring module from the coherent receivers and saved in the SDON data repository, by using a service such as the one presented in [33]. We do not consider here the OPM data possibly collected in ROADM switching nodes because we focus on the lowest-cost monitoring scheme that excludes expensive optical spectrum analyzers in ROADMs. To perform this step, SDON controllers typically adopt standardized interfaces and protocols such as NETCONF. Once new OPM data is collected and properly stored, the security assessment module is triggered to perform the appropriate processing, retrieving the OPM data from the data repository. At this stage, it is desirable that the security assessment module completes data processing within the monitoring cycle (e.g., one minute), i.e., before the next monitoring data arrives.

The following data preprocessing step removes outliers (e.g., due to transients during channel addition) and normalizes the features. Since OPM data can encompass very different parameters with large differences in their values and scales, data normalization plays a crucial role in easing the processing and learning by the ML algorithms. The normalization is done for each of the features individually, and can be implemented using several techniques such as z-score standardization.

The next step processes the normalized data to perform attack detection and identification. The main objective of this step is to detect whether any of the evaluated lightpaths is affected by an attack and identify the type and intensity of the attack. The goal of the framework is to classify attacks with minimal false negative and false positive rates. Note, however, that false negatives represent a higher risk for network operation than false positives in the context of attack detection and identification. For example, false negatives lead to attacks remaining undetected, potentially evolving to more disruptive events, while false positives could trigger unnecessary countermeasures such as protection rerouting, causing an overhead in resource usage, but do not bring as high security risk as the false negatives. There are several algorithms that can be used to perform the classification needed to solve the attack detection and identification problem, and the most promising ones are detailed in the following.

B. Machine-Learning-Aided Attack Classification

The attack detection and identification problem can be formulated as a classification problem, which falls within the supervised learning type of ML problems. In a classification problem, a classifier receives a sample as input, and outputs the class to which that sample belongs. While training time requirements may vary depending on the deployment scenario, an important requirement is that the processing time for the algorithm should be shorter than the monitoring cycle, i.e., the time between two consecutive monitoring measurements. There is an ample number of algorithms that can be used to perform classification. Although there is prior literature analyzing the performance of classifiers over a number of problems, it is difficult to infer which classifier is more suitable for an unseen problem, such as the one we are analyzing in this paper. Moreover, several algorithms, such as the ones described below, have complexity that increases linearly with the number of features and samples, demonstrating potential efficiency for the ADI, except for the decision tree, whose complexity increases quadratically with the number of features [30]. Therefore, we consider a number of classifiers with different characteristics, in an effort to assess their benefits and drawbacks when applied to the particular problem tackled in this paper. We refer to [34] for an in-depth assessment of classifiers' performance. In the following, we briefly introduce the classifiers considered in this work.

1) Artificial Neural Network (ANN): mimics the human nervous system by forming layers of artificial neurons that communicate with each other via weighted combinations of input and output function values. It is able to learn complex relations between inputs and outputs, as well as to process complex data such as images.

2) Support Vector Machine (SVM): uses a spatial model of the data, mapped so as to achieve gaps as wide as possible between different classes. The kernels used by SVM can transform the data to enable better spatial separation and clearer interpretation of the classifier.

3) Gaussian Process (GP): implements a Gaussian process for probabilistic classification. This type of classifier can

handle multi-class classification (which is the one suitable for our problem) by extending the Gaussian process on a per-class or per-pair-of-classes basis. It can capture model uncertainties and be tuned with prior knowledge of the problem.

4) Decision Tree (DT): classifies the samples by learning simple decision rules inferred from the training data. An important parameter of a decision tree is its depth, which defines how complex it is, and may also help in improving accuracy. An advantage of DTs is that their representation can be easily interpreted, and implementation can be done using simple computer language control instructions.

5) Random Forests (RF): use a number of decision trees fitted on subsamples of the training data. The results from the decision trees are averaged to improve accuracy and prevent overfitting. It has shown good performance for a variety of datasets [34].

6) Naive Bayes (NB): implements a probabilistic classification algorithm. This classifier applies Bayes' theorem with strong (naive) independence assumption between the features. If this assumption holds, NB can converge quickly or needs less training data.

7) Nearest Neighbors (NN): different from other models, k-nearest neighbors is instance-based learning. It does not learn a model out of the training data, but instead saves the data for future queries. Thus, for each new sample, the algorithm computes the class based on the most representative class out of k neighbors. The learning process is quite fast, as it saves all the training data for further use.

8) Quadratic Discriminant Analysis (QDA): is a classifier that builds a quadratic decision surface in order to differentiate the classes. It is derived from probabilistic models which model the class conditional distribution of the data for each class.

IV. Experimental Testbed and Measurement Procedures

The experiment is designed to emulate a link of a transport network affected by either an intrusion signal injected along the fiber line, or by high speed polarization modulation. A meshed network scenario is not considered here for the lack of ROADM nodes in the present testbed. The experimental setup is shown in Fig. 3.

The optical signal under test is a 200 Gbit/s polarization multiplexed 16QAM signal generated by a commercial transponder (Coriant Groove G30). The line system is loaded with 10 additional Continuous Wave (CW) optical channels, whose wavelengths range from 1541.4 to 1558.2 nm, to emulate realistic operating conditions. Since transmission channel crosstalk is not relevant in this work, the lack of modulation on the CW loading signals does not jeopardize the final results.

These optical channels are aggregated by a passive multiplexer and sent to one of the input ports of a Lumentum Transport ROADM Whitebox/Graybox which encompasses all the functions of a ROADM line interface: 1x20 flexgrid WSS, optical amplifiers, monitoring devices, board controller, etc. The Dense Wavelength Division

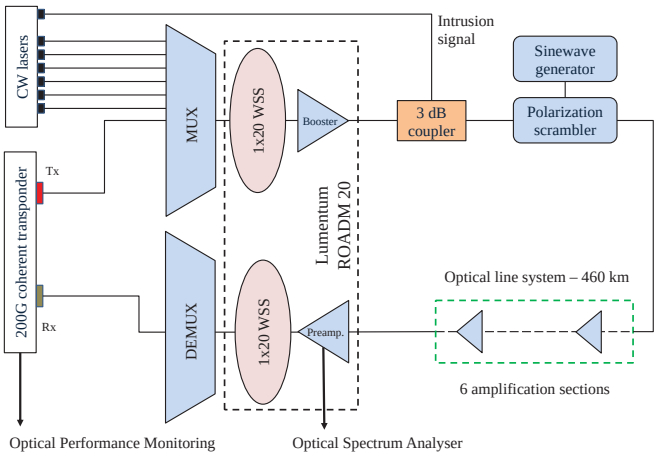


Fig 3. Experimental setup used in the network attack experiment.

Multiplexing (DWDM) signal exiting the booster amplifier is then combined with a CW intrusion signal by a 3 dB coupler emulating a bent-fiber temporary coupler that might be used for a field attack. The signal then passes through an all-fiber polarization modulator composed of 3 piezoelectric fiber squeezers (General Photonics, PolaRITE-II) which implements the polarization attack (details of attack scenarios are given in the following subsections). Just one of the fiber squeezers is used; it is driven by a sinewave signal at 110 kHz frequency, which corresponds to one of the resonant frequencies of the piezoelectric element and therefore produces a deep polarization modulation. The polarization modulator loss is negligible, and it does not perturb the DWDM signal except when the sinewave generator is switched on to simulate a polarization attack.

Finally, the signal is delivered to the optical line system which includes 6 field amplification sections, each composed of an optical line amplifier and a 76 km span of G.652 fiber. The line amplifiers work in the transparency regime, i.e., the gain compensates for the fiber span loss, and the launch power is set at 0 dBm per channel in all sections.

At the output of the optical line system, the signal is sent to the line input of the Lumentum ROADM and finally delivered to the transponder receiver after passing through a passive

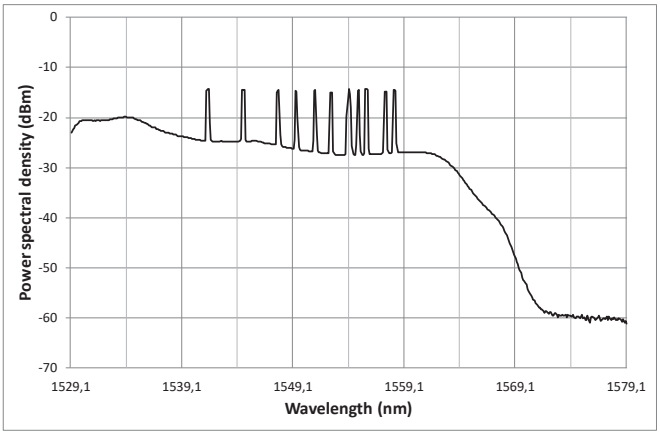


Fig 4. Power spectrum of the baseline DWDM signal at the preamplifier monitoring port. The OSA resolution is 0.5 nm.

demultiplexer. An Optical Spectrum Analyzer (OSA) is connected to the monitor port of the preamplifier to measure the spectra of the received DWDM signal.

The power spectrum of the baseline DWDM signal at the preamplifier monitoring port is shown in Fig. 4. The power of the 11 optical channels has been manually equalized within ± 2 dB by changing their launch power. The system in the normal working condition, i.e., without any intrusion signal or polarization modulation, operates error-free with 20.6 dB $OSNR_{0.1}$ (measured with 0.5 nm resolution and rescaled to 0.1 nm). Considering the Optical Signal-to-Noise Ratio (OSNR) sensitivity limit of the Groove G30 transponder, the system operates with a 2.6 dB OSNR margin which can be considered representative of real field conditions. While many other operative conditions with different OSNR and BER values may be implemented as normal working statuses, we have selected the described one because it is close enough to the transmission limit of the Groove G30 transponder to make it quite vulnerable to attacks. The OPM data provided by the coherent receiver are downloaded in 10 seconds intervals by an application based on the NETCONF protocol. The OPM parameters collected by the system are shown in Table I. A first set of OPM data with 400 samples (acquired during 67 minutes) was collected in this condition and automatically labelled as the baseline scenario of our experiment.

A. *In-Band Jamming Attack*

In the in-band jamming attack, the intrusion signal is a CW low power signal whose frequency falls within the bandwidth of the signal under test as shown in Fig. 5. We have experimentally assessed that when the intrusion signal is slightly detuned with respect to the central frequency of the signal under test, the jamming is particularly effective (i.e. a remarkable increase in BER-POST-FEC can be achieved by modest intrusion signal power). For this reason, we detuned the two signals by about 10 GHz as shown in Fig. 5.

We have emulated two in-band jamming attack conditions, i.e., light and strong, by setting the power of the intrusion signal

TABLE I
OPTICAL PERFORMANCE MONITORING (OPM) PARAMETERS CONTAINED IN EACH DATA SAMPLE

Acronym	Description
CD	Chromatic Dispersion
DGD	Differential Group Delay
OSNR	Optical Signal to Noise Ratio
PDL	Polarization Dependent Loss
Q-factor	Q factor
BE-FEC	Block Errors before FEC
BER-FEC	Bit Error Rate before FEC
UBE-FEC	Uncorrected Block
BER-POST-FEC	Bit Error Rate after FEC
OPR	Optical Power Received
OFR	Optical Frequency Received
LOS	Loss Of Signal

For all parameters except BE-FEC, UBE-FEC, and LOS, the system provides the maximum, minimum and average values in the observation interval.

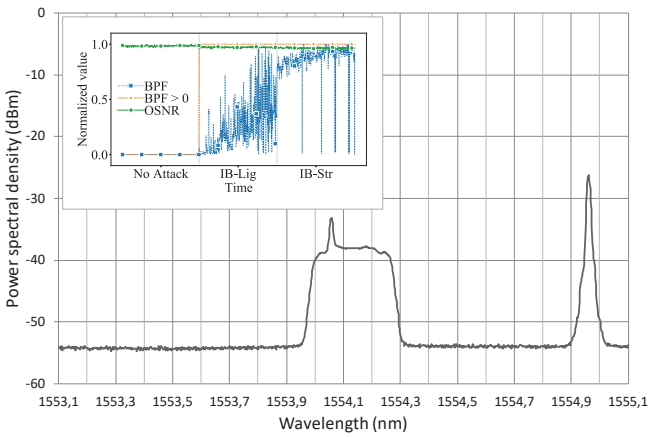


Fig 5. Power spectrum of the channel under test in the in-band (IB) jamming attack scheme (-12 dB intrusion signal relative power). The spectrum is measured with maximum OSA resolution (0.01 nm). The peak on the righthand side of the picture is one of the CW loading signals. The inset shows the normalized OSNR and BER-POST-FEC (BPF) for the light and strong attack intensity. The presence of uncorrected errors is highlighted by BPF>0.

to 14 dB and 12 dB below the power of the signal under test, respectively. In these conditions, the system produces Uncorrected Block Errors (UBE-FEC) errors ranging from a few errors to many thousand errors per minute for the light and strong attack condition, respectively. A full OPM dataset with 400 samples has been collected for each condition.

B. *Out-of-Band Jamming Attack*

In the out-of-band jamming attack, the intrusion signal is a CW signal with a frequency outside the bandwidth of the signal under test as shown in Fig. 6. The intrusion signal at 1562.2 nm wavelength was generated by one of the available CW lasers. In this type of attack, the power required for the intrusion signal is much higher than for the in-band attack. In this case, the intrusion signal causes power reduction of the other channels, provided that its power is of the same order of magnitude as the other signals (typically 0 to 3 dBm). This in turn produces a reduction of the OSNR that impairs the working channels and degrades their performance. We have

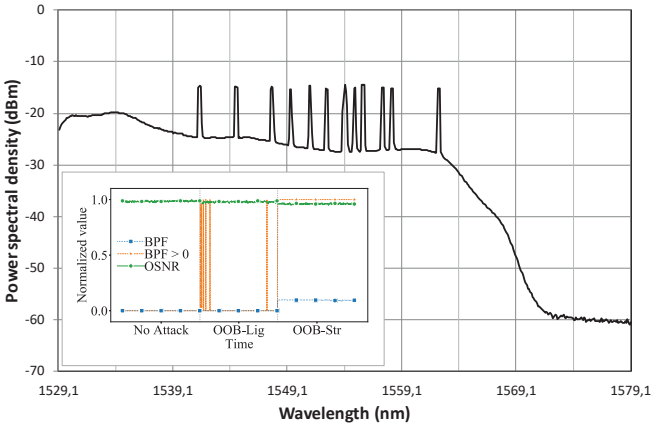


Fig 6. Power spectrum of all channels in the out-of-band (OOB) jamming attack scheme (0 dBm intrusion signal relative power). The intrusion signal is the one on the righthand side of the spectrum (1562.2 nm) and has the same power as the optical channel under test. The inset shows the normalized OSNR and BER-POST-FEC (BPF) for the light and strong attack intensity. The presence of uncorrected errors is highlighted by BPF>0.

implemented two out-of-band jamming attack conditions, i.e., light and strong, by setting the power of the intrusion signal to 0 dB and +3 dB with respect to the power of the signal under test, respectively. In these conditions, the system produces UBE-FEC errors ranging from a few errors to many thousand errors per minute for the light and strong attack condition, respectively. A full OPM dataset with 400 samples has been collected for each condition.

C. Polarization Modulation Attack

In the polarization attack, we have switched off the intrusion signals and activated the polarization state modulator. In this operating condition, all optical parameters of the system are the same as in the baseline condition, but the polarization modulation causes transmission errors as soon as it is faster than the coherent receiver polarization recovery algorithm [35].

We have experimentally identified one of the resonant frequencies of the fiber squeezer at 110 kHz by monitoring the amplitude of the sinewave driving signal. Then, the sinewave amplitude was set to 0.14 and 0.4 V peak-to-peak resulting in light and strong attack, respectively. In these conditions, the system produces UBE-FEC errors ranging from a few errors to many thousand errors per minute for the light and strong attack condition, respectively. A full OPM dataset with 400 samples has been collected for each attack condition.

V. Results

The framework described in Sec. III was used to analyze the real-world attack scenarios from the experimental setup detailed in Sec. IV. The experiment collected 400 samples for each attack scenario of interest, yielding a total of 2800 samples across the 7 scenarios, and generating a balanced dataset, where each attack scenario is represented by the same number of samples in the dataset. Each sample contains a total of 30 parameter values, described in Table I. Out of the dataset, 50% of the samples were used for training the ML classifiers, 10% were used for validation (in the classifiers that allow for it), and the remaining 40% were used for testing. We also performed *k*-fold cross-validation to confirm that the dataset split is not biasing the results. The *k*-fold cross-validation splits the dataset into *k* folds. Then, it trains the model for each combination of *k*-1 folds, while validating the trained model with the remaining 1 unseen fold. After repeating this process for every combination of folds, the average and standard deviation of the accuracy obtained for the unseen folds is used to evaluate the reliability of the classifier. Models which obtain high accuracy and low standard deviation are regarded as reliable.

The framework was implemented using Python[1]. Data retrieval and outlier removal was implemented using Pandas [36]. Data normalization and classification used the implementation provided by Scikit-learn [37]. In particular, z-score standardization technique was used for data normalization, by computing the average and the standard deviation for each feature and representing each data point by the multiple of standard deviations from the average. The performance of the classifiers described in Sec. III-B was evaluated for the attack classification problem required in the ADI framework. For all classifiers, the 30 OPM parameters described in Table I are used as input.

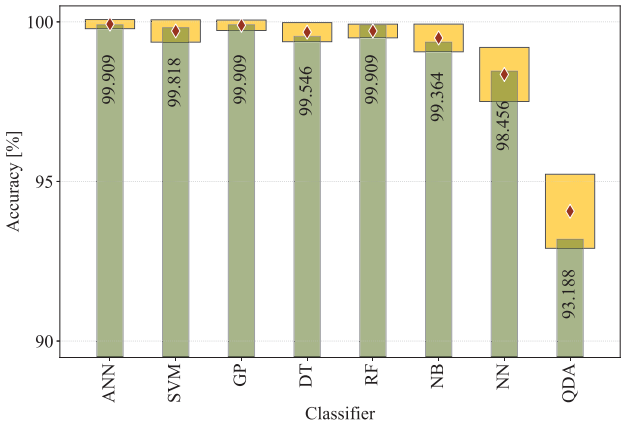
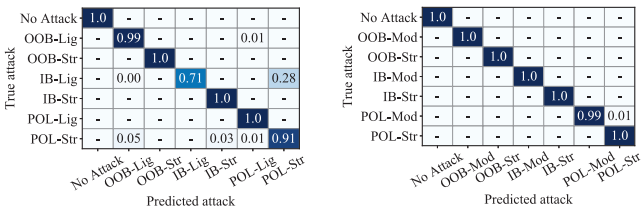


Fig. 7. Test and cross-validation accuracy of the considered ML classifiers. Bars represent the average test accuracy for the 50% test dataset. Diamond marker shows the average cross-validation accuracy, while interval bars show the standard deviation of the accuracy over the 10 folds.

Some classifiers provide the means to configure some of their parameters. We tested several classifier configurations and report the one that yielded the best classification accuracy. For the ANN, two hidden layers were used, with 50 and 100 neurons, using linear and tanh activation functions, respectively. Initialization of weights was performed using the Xavier procedure [38]. The output layer was composed of 7 neurons using the softmax activation function, appropriate for classification purposes. The training was done over 1000 iterations using Adam optimizer with learning rate of 0.0001, configured to optimize classification accuracy. Nearest neighbors used k=5. Decision tree and random forest were configured for maximum depth equal to 5, with random forest having 10 estimators. The other parameters were kept according to the defaults set by Scikit-learn.

In our performance assessment, we first evaluate the classification accuracy of the different classifiers. We then analyze the distribution of the classification error over the different classes for two representative classifiers (the ones achieving the worst and the best performance). Furthermore, we assess the OPM parameters that contribute the most to the accuracy of the classifiers, with the goal of finding the smaller set of parameters that can provide the maximum accuracy. We also investigate classifier accuracy for datasets with missing features to analyze their robustness to OPM parameter set incompleteness. Finally, we assess the execution time for the classification of a number of samples, in order to evaluate the maximum number of lightpaths that can be monitored per ADI instance.

[1] The implementation is available at https://github.com/carlosnatalino/JLT-2019-Experimental-ML-attacks. The data is not shared due to confidentiality.



(a) QDA (b) ANN

Fig. 8. Confusion matrices for the worst (QDA) and the best (ANN) classifier.

A. Classification Accuracy

Fig. 7 shows the classification accuracy of the classifiers for the test set, i.e., unseen monitoring samples, and for the cross-validation using 10 folds. We define accuracy as the percentage of correctly classified samples. All considered classifiers achieve accuracy higher than 90%. The ANN, GP and RF achieve the same highest test accuracy, correctly classifying more than 99.9% of samples. However, in cross-validation, the ANN is the only classifier to achieve 99.9% accuracy on average, and the one presenting the lowest standard deviation. This is explained by the capacity of ANNs to accurately model intricate relations between the input and the output, even for very complex phenomena. QDA exhibits the poorest performance, achieving only 93% accuracy, while having the highest standard deviation of the cross-validation.

The above results for the ANN indicate that the model, tested here for the single-hop scenario, has good generalization power to identify attacks through holistic assessment of the OPM parameter set collected at the coherent receivers. Therefore, it can be expected (with some hyperparameters fine-tuning) to perform well even in multi-hop scenarios, where the optical connections may be impacted in other ways by co-propagating connections and traversed devices, potentially posing a more challenging scenario for the ML algorithms.

For the problem investigated in this paper, accuracy alone does not fully characterize the performance of a classifier. Since false negatives are potentially more hazardous than false positives when it comes to security breach detection (as explained in Sec. III-A), the confusion matrix may further help to identify the best performance. Fig. 8 presents the confusion matrices obtained by ANN and QDA as the best and the worst classifier, respectively, for the considered test set. Confusion matrices are used to assess classification accuracy in a per-class manner (i.e., per-attack scenario, in our case). In general classification problems, the objective is to concentrate the outcomes of the classifier along the main diagonal of the matrix. In addition, our scenario also calls for a false-negative rate (depicted in the left-most column) as low as possible.

QDA attains the lowest classification accuracy, which translates into a confusion matrix with several misclassifications (Fig. 8a). More specifically, 29% of in-band attacks are misclassified as either polarization (>28%) or out of band (<1%) attacks, while 8% of the polarization attacks are also misclassified either as in- (5%) or out-of-band (3%) attacks. This means that QDA misinterprets attack types, which can lead to triggering inappropriate recovery strategies. The

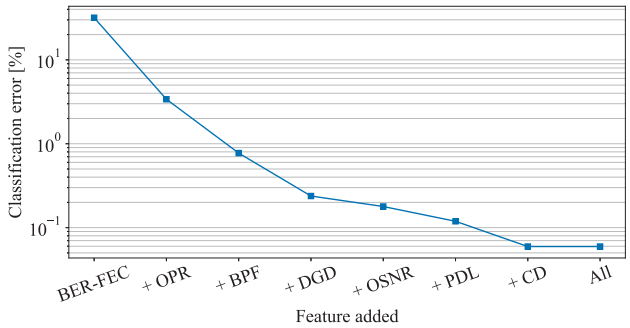


Fig. 9. Classification error for ANN starting with one feature, and sequentially adding the feature that yields the lowest error (BER-POST-FEC is denoted by BPF).

ANN confusion matrix (Fig. 8b) indicates the highest classification accuracy. The only misclassification observed for the ANN is related to the 1% of samples of a light polarization attack classified as a strong polarization attack. Such cases do not represent a significant danger for network operation, as effective measures to counteract the attack will be triggered even for the misclassified cases.

B. Key Features for Classification Accuracy

In addition to the performance assessment considering the entire dataset, we assess the significance of each of the OPM parameters from Table I. The ANN classifier was used to perform this assessment. To this end, we start by evaluating the accuracy performance by having only one of the OPM parameters from Table I in the training/test dataset. After identifying the parameter that provides the highest accuracy, we permanently add it to the dataset, and repeat the process for the remaining parameters.

Fig. 9 presents the error (the inverse of the accuracy) observed by following the procedure just described. The error is showed instead of accuracy for better visualization of the contribution of each added parameter. For a dataset with a single OPM parameter, BER-FEC results in the lowest error (i.e., 32%). In the next iteration where another of the remaining parameters is added to the dataset holding BER-FEC, OPR arises as the parameter that attains the strongest error reduction.

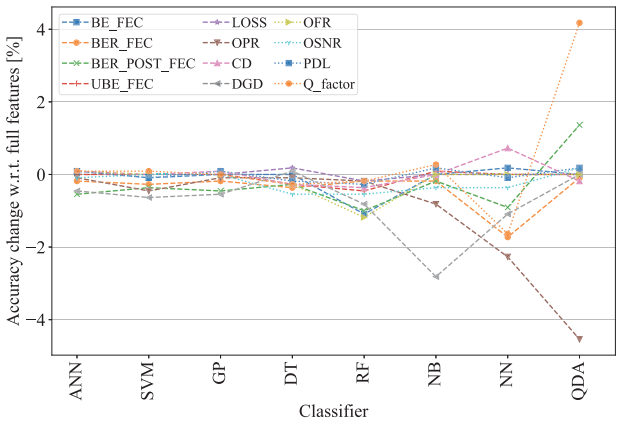


Fig. 10. Accuracy change when features are missing in comparison to the full feature case, i.e., the difference between accuracy with missing features and accuracy with full features, relative to the latter.

In this case, ANN already reaches more than 90% accuracy, i.e., the error diminishes from 32% with BER-FEC only to 3.4% with BER-FEC and OPR. The third parameter that contributes the most to the decrease in classification error is the BER-POST-FEC. When the dataset contains BER-FEC, OPR and BER-POST-FEC, the ANN obtains an error lower than 1%, i.e., the accuracy increases from 96.6% to 99.2%. In the subsequent iterations, DGD, OSNR and PDL are the features that reduce the error the most. Finally, after adding CD, the ANN achieves its maximal accuracy, which is higher than 99.9%.

The results show that by considering 7 out of the 12 collected OPM parameters described in Table I, the highest performance observed for the ANN classifier can be achieved. Reducing the number of considered OPM parameters can potentially reduce complexity during training. Shorter training times may enable network operators to update their models more quickly if a new kind of attack is discovered. Similar procedures can be used by other ML algorithms for identifying key parameters for specific network tasks.

C. Classification Accuracy for Datasets with Missing OPM Parameters

In addition to the performance assessment considering the entire dataset, and the one to find the most significant OPM parameters, we assess the accuracy change observed for each classifier in case of removal of each individual OPM parameter. To this end, for each OPM parameter in Table I, we generated a new dataset where the parameter of interest is set to zero in all samples.

Fig. 10 presents the accuracy change observed for each classifier when a given OPM parameter is missing. This change is defined as the difference between the accuracy achieved for the set without the missing feature and the accuracy achieved for the full set of features, relative to the latter. It can be noticed that the removal of the key parameters identified in the previous section (e.g., BER-FEC, OPR, BER-POST-FEC) causes the highest losses in accuracy. However, some classifiers, such as ANN, GP and DT, are minimally impacted by the missing features. On the contrary, NN and QDA are significantly impacted by the missing features. In particular, the accuracy of QDA drops by more than 4% when OPR is removed from the dataset. Interestingly, QDA is also the most positively impacted by missing features. For instance, when BER-FEC is removed from the dataset, QDA improves its classification accuracy by up to 4%. However, this 4% improvement in performance would result in 98% overall accuracy, which remains below the accuracy of most of the classifiers tested.

D. Scalability of the Classifiers

In the last part of this work, we assess how many lightpaths would be possible to monitor by the ADI framework. For this purpose, we measure the time required by each classifier to classify up to one million randomly generated samples. For this assessment, we use a Red Hat Enterprise Linux workstation with an Intel Xeon CPU E5-1660 v3 with 8 cores and 16 threads clocked at 3.00GHz and 64 GB of RAM. The platform uses Python 3.6 and Scikit-learn 0.20.0. All classifiers use their CPU implementation, but some of them could benefit from running

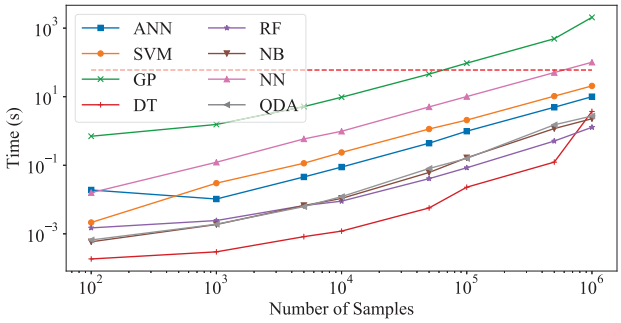


Fig. 11. Execution time of the classifiers for up to 1 million samples (i.e., lightpaths). Red dashed line represents the 1-minute threshold.

in GPUs to boost speed [39]. To eliminate the impact of the dataset on the performance, we randomly generate a synthetic dataset with the desired number of samples using a normal distribution defined, for each feature, by its mean and standard deviation computed from the experimentally generated dataset.

Fig. 11 shows the time required by each classifier to classify a given number of samples, with the dashed line denoting the 1-minute threshold, as recommended in [40]. The trend shown in the figure confirms that the complexity of the algorithms is linear with respect to the number of samples. Considering that monitoring data are collected from the network every minute, we can see that most of the classifiers are capable of processing up to one million lightpaths during the monitoring cycle. In particular, ANN is capable of classifying the one million samples in less than 10 seconds, leaving space for other operations that might be necessary in the monitoring platform, or for a shorter monitoring cycle.

VI. Conclusion

This paper experimentally evaluates a machine-learning-aided framework for physical-layer attack detection and identification over a range of attack techniques with varying intensities. A set of 8 ML classifiers is evaluated, revealing that ANN achieves the highest classification accuracy (higher than 99.9%), whereas no false negatives are observed.

The obtained results show the significance of the OPM parameters to accurate attack detection by the ANN. It is possible to identify 7 OPM parameters which enable the ANN to achieve the same accuracy as when it uses all 12 collected OPM parameters. We also evaluated datasets with missing features. The results show that ANN performs well even in cases where any single OPM parameter is unavailable. For some ML classifiers, e.g., QDA, limiting the input features may increase accuracy by up to 4%. This emphasizes the need for careful selection of the OPM data to be input to the deployed models so as to enhance classifier performance.

The scalability assessment indicates that most of the classifiers are able to classify a million samples (i.e., a million lightpaths) in less than one minute, using a standard off-the-shelf workstation. Moreover, ANN is able to classify a million lightpaths in less than 10 seconds, leaving a sufficient portion of the 1-minute monitoring cycle for other tasks.

Although the framework presents good performance, its deployment in real-world infrastructures remains an open task.

The integration of the framework with state-of-the-art monitoring or SDON controller software is also an interesting challenge. The framework for cognitive security diagnostics of individual connections presented in this paper creates a foundation for detecting attack types and techniques among the multitude of connections in the network and exhibits strong potential to enhance the level of optical network security.

REFERENCES

[1] N. Skorin-Kapov *et al.*, "Physical-layer security in evolving optical networks," *IEEE Commun. Mag.*, vol. 54, no. 8, pp. 110–117, Aug. 2016.
[2] D. FitzGerald, "FBI investigates new attack on internet fiber optic cables," *Wall Street J.*, June 2015.
[3] InfoGuard, "Data security in the converged enterprise network," White paper, Dec. 2018.
[4] T. Uematsu *et al.*, "Design of a temporary optical coupler using fiber bending for traffic monitoring," *IEEE Photonics J.*, vol. 9, no. 6, pp. 1–13, Dec. 2017.
[5] J. Mata *et al.*, "Artificial intelligence (AI) methods in optical networks: A comprehensive survey," *Opt. Switching Netw.*, vol. 28, pp. 43–57, Apr. 2018.
[6] F. Musumeci *et al.*, "An overview on application of machine learning techniques in optical networks," *IEEE Commun. Surveys Tuts.*, pp. 1–1, Dec 2018.
[7] S. Shahkarami *et al.*, "Machine-learning-based soft-failure detection and identification in optical networks," in *Proc. of OFC*, 2018, p. M3A.5.
[8] A. P. Vela *et al.*, "Soft failure localization during commissioning testing and lightpath operation," *IEEE/OSA J. Opt. Commun. Netw.*, vol. 10, no. 1, pp. A27–A36, Jan. 2018.
[9] D. Rafique *et al.*, "Cognitive assurance architecture for optical network fault management," *IEEE/OSA J. Lightw. Technol.*, vol. 36, no. 7, pp. 1443–1450, Apr. 2018.
[10] C. Natalino *et al.*, "Field demonstration of machine-learning-aided detection and identification of jamming attacks in optical networks," in *Proc. ECOC*, 2018, p. We2.58.
[11] M. Fok *et al.*, "Optical layer security in fiber-optic networks," *IEEE Inf. Foren. Sec.*, vol. 6, no. 3, pp. 725–736, Apr. 2011.
[12] M. Furdek, "Towards secure and self-diagnosable optical networks [invited]," in *Proc. PISC*, 2018, pp. Fr3A.1.1–3.
[13] T. Peng *et al.*, "Propagation of all-optical crosstalk attack in transparent optical networks," *Opt. Eng.*, vol. 50, no. 8, pp. 085 002.1–3, Aug 2011.
[14] N. Skorin-Kapov *et al.*, "A new approach to optical networks security: Attack-aware routing and wavelength assignment," *IEEE Trans. Netw.*, vol. 18, no. 3, pp. 750–760, June 2010.
[15] N. Skorin-Kapov *et al.*, "Wavelength assignment for reducing in-band crosstalk attack propagation in optical networks: ILP formulations and heuristic algorithms," *European J. Oper. Res.*, vol. 222, no. 3, pp. 418–429, Nov 2012.
[16] Z. Zhu *et al.*, "Build to tenants' requirements: On-demand application-driven vSD-EON slicing," *IEEE/OSA J. Opt. Commun. Netw.*, vol. 10, no. 2, pp. A206–A215, Feb. 2018.
[17] M. Furdek *et al.*, "Attack-aware dedicated path protection in optical networks," *IEEE/OSA J. Lightwave Techn.*, vol. 34, no. 4, pp. 1050– 1061, Feb. 2016.
[18] E. Hugues-Salas *et al.*, "Experimental demonstration of DDoS mitigation over a quantum key distribution (QKD) network using software defined networking (SDN)," in *Proc. OFC*, 2018, p. M2A.6.
[19] R. Rejeb *et al.*, "Multiple attack localization and identification in all-optical networks," *Opt. Switching Netw.*, vol. 3, no. 1, pp. 41–49, July 2006.
[20] C. Mas *et al.*, "Failure location algorithm for transparent optical networks," *IEEE J. Sel. Areas Commun.*, vol. 23, no. 8, pp. 1508–1519, Aug. 2005.
[21] D. Monoyios *et al.*, "Attack-aware resource planning and sparse monitor placement in optical networks," *Opt. Switching Netw.*, vol. 29, pp. 46–56, July 2018.
[22] F. Pederzolli *et al.*, "Towards secure optical networks: A framework to aid localization of harmful connections," in *Proc. OFC*, 2018, p. Th2A.42.
[23] R. M. Morais and J. Pedro, "Machine learning models for estimating quality of transmission in DWDM networks," *IEEE/OSA J. Opt. Commun. Netw.*, vol. 10, no. 10, pp. D84–D99, Oct. 2018.
[24] R. Proietti *et al.*, "Experimental demonstration of machine-learning- aided QoT estimation in multi-domain elastic optical networks with alien wavelengths," *IEEE/OSA J. Opt. Commun. Netw.*, vol. 11, no. 1, pp. A1–A10, Jan. 2019.
[25] T. Tanimura *et al.*, "Convolutional neural network-based optical performance monitoring for optical transport networks," *IEEE/OSA J. Opt. Commun. Netw.*, vol. 11, no. 1, pp. A52–A59, Jan. 2019.
[26] Y. Ou *et al.*, "Field-trial of machine learning-assisted quantum key distribution (QKD) networking with SDN," in *Proc. ECOC*, 2018, p. Mo3D.3.
[27] L. Velasco *et al.*, "Learning from the optical spectrum: Soft-failure identification and localization [invited]," in *Proc. OFC*, 2018, p. W1G.1.
[28] X. Chen *et al.*, "On real-time and self-taught anomaly detection in optical networks using hybrid unsupervised/supervised learning," in *Proc. ECOC*, Sept 2018, p. We1D.4.
[29] Y. Li *et al.*, "Light source and trail recognition via optical spectrum feature analysis for optical network security," *IEEE Commun. Lett.*, vol. 22, no. 5, pp. 982-985, May 2018.
[30] M. Bensalem *et al.*, "Machine learning techniques to detecting and preventing jamming attacks in optical networks," arXiv: 1902.07537v1[cs.N1] 20 Feb 2019 arxiv.org/pdf/1902.07537.pdf
[31] S. Yan *et al.*, "Multilayer network analytics with SDN-based monitoring framework," *IEEE/OSA J. Opt. Commun. Netw.*, vol. 9, no. 2, pp. A271–A279, Feb. 2017.
[32] A. S. Thyagaturu *et al.*, "Software defined optical networks (SDONs): A comprehensive survey," *IEEE Commun. Surveys Tuts.*, vol. 18, no. 4, pp. 2738–2786, 4th quarter 2016.
[33] F. Paolucci *et al.*, "Network telemetry streaming services in SDN-based disaggregated optical networks," *IEEE/OSA J. Lightwave Techn.*, vol. 36, no. 15, pp. 3142–3149, Aug. 2018.
[34] M. Fernandez-Delgado *et al.*, "Do we need hundreds of classifiers to solve real world classification problems?" *J. Mach. Learn. Res.*, vol. 15, pp. 3133–3181, Jan. 2014.
[35] P. M. Krummrich *et al.*, "Demanding response time requirements on coherent receivers due to fast polarization rotations caused by lightning events," *Opt. Express*, vol. 24, no. 11, pp. 12 442–12 457, May 2016.
[36] W. McKinney, "Data structures for statistical computing in Python," in *Proc. SciPy*, 2010, pp. 51 – 56.
[37] F. Pedregosa *et al.*, "Scikit-learn: Machine learning in Python," *J. Mach. Learn. Res.*, vol. 12, pp. 2825–2830, 2011.
[38] X. Glorot and Y. Bengio, "Understanding the difficulty of training deep feedforward neural networks," in *Proc. AIStat*, 2010, pp. 249–256.
[39] Z. Wen et al., "ThunderSVM: A fast SVM library on GPUs and CPUs," *J. Mach. Learn. Res.*, vol. 19, no. 21, pp. 1–5, 2018.
[40] A. P. Vela *et al.*, "Distributing data analytics for efficient multiple traffic anomalies detection," *Comput. Commun.*, vol. 107, pp. 1 – 12, July 2017.