Adversarial Analysis of ML-based Anomaly Detection in Multi-layer Network Automation

Xiaoqin Pan, Hao Yang, Zichen Xu, and Zuqing Zhu, *Senior Member, IEEE*

***Abstract*—The fast development of multi-layer packet-over-optical networks has made network monitoring and troubleshooting increasingly complicated. This has stimulated people to combine machine learning (ML) and software-defined networking (SDN) to realize multi-layer network automation. Despite its initial successes, the vulnerabilities of multi-layer network automation have not been fully explored. This work studies how to mislead the ML-based classifiers for anomaly detection. Specifically, we design two adversarial-sample-based attack schemes based on the white-box attack (WBA) and black-box attack (BBA) strategies, respectively, to eavesdrop and tamper legitimate telemetry data samples and generate adversarial samples adaptively, for disturbing ML-based classifiers and in turn misleading network automation to make incorrect decisions. Compared with WBA, BBA makes the attack scheme more practical by minimizing the dependency on pre-knowledge of the target ML-based classifiers. Considering different types of ML-based classifiers, we build a real-world packet-over-optical testbed and leverage the telemetry samples collected in it to demonstrate that our proposed BBA scheme can interact with the network quietly to train itself, generate well-crafted adversarial samples to tamper legitimate telemetry samples in the hard-to-detect way, and mislead ML-based classifiers in the network automation system to severely affect their performance on anomaly detection.**

***Index Terms*—Artificial intelligence (AI), Machine learning (ML), Network automation, Vector homomorphic encryption (VHE), Adversarial samples, White box attack, Black box attack.**

I. Introduction

RECENTLY, with the rise of emerging services such as 5G, Big Data and cloud computing, network systems have become increasingly complex and highly dynamic [1], raising new challenges for network control and management (NC&M). Meanwhile, the rapid deployment of virtualization technologies (*e.g.*, virtual network slicing [2, 3] and network function virtualization (NFV) [4, 5]) has reshaped the Internet more flexible at the cost of making it more difficult to detect and locate network faults/anomalies. Moreover, to ensure the quality-of-service (QoS) of various network services, the multi-layer architecture of metro/core networks (*i.e.*, packet-over-optical) requires NC&M to make intelligent and timely decisions [6]. These issues added up to stressing the research and development (R&D) of NC&M over time, especially for that of multi-layer packet-over-optical networks.

X. Pan, H. Yang, Z. Xu, and Z. Zhu are with the School of Information Science and Technology, University of Science and Technology of China, Hefei, Anhui 230027, China (email: zqzhu@ieee.org).

X. Pan is also with the Engineering Technology Center, Southwest University of Science and Technology, Mianyang, Sichuan 621010, China.

H. Yang is also with the School of Information Engineering, Southwest University of Science and Technology, Mianyang, Sichuan 621010, China.

Manuscript received on January 10, 2022.

The symbiosis of software-defined networking (SDN) [7] and machine learning (ML) opened up new opportunities for NC&M. Specifically, with the centralized control provided by SDN and the "observe-analyze-act" decision loop facilitated by ML, network automation can be realized for various types of networks, including packet-over-optical networks [8]. Fig. 1 shows an example of network automation in a packet-over-optical network. With the help of in-band network telemetry (INT) [9], real-time status of network elements in the packet and optical layers can be collected in a fine-grained manner to visualize the dynamic network environment (*Observe*) [10]. Then, the telemetry data is analyzed by the ML-based data analytics (DA) module to not only extract global information about the network but also detect anomalies accurately (*Analyze*) [11]. Finally, the analysis results are forwarded to the SDN controller for it to make timely NC&M decisions and update network configuration to adapt to the new network state, *e.g.*, a newly-generated anomaly can be addressed with the proper fault recovery mechanism (*Act*) [12]. Hence, network automation can make NC&M more effective and minimize unnecessary human interventions during network operation.

Although network automation is promising, we should always be cautious about security issues when reducing human interventions in NC&M. The operation of ML modules can be disturbed with data poisoning, which refers to a malicious party misleading an ML module by sneaking well-craft adversarial samples in its inputs [13]. The adversarial-sample-based attacks can be launched in a number of ways. For instance, the attacker can eavesdrop the data reporting channels between control and data planes for legitimate telemetry data samples, and then generate adversarial samples based on them to inject back in the channels [14]. This is feasible because the commonly-used protocols for setting up secure data reporting channels (*e.g.*, the transport layer security (TLS)) is vulnerable to the man-in-the-middle attack [15]. On the other hand, due to the shortage of the expertise, hardware/software resources and labor for designing and training sophisticated ML modules, an operator might leverage "machine-learning-as-a-service" (MLaaS) [16] to outsource its ML module to a third-party entity. However, MLaaS is vulnerable to data poisoning, as an attacker can hack into the MLaaS system to contaminate ML modules quietly with adversarial samples [17].

Therefore, it is relevant to conduct adversarial analysis of the ML modules that can be used in network automation systems to check how secure they are. Nevertheless, most of the existing studies on network automation in optical or packet-over-optical networks did not pursue the research in this direction. Previously, in [18], we showed that the ML

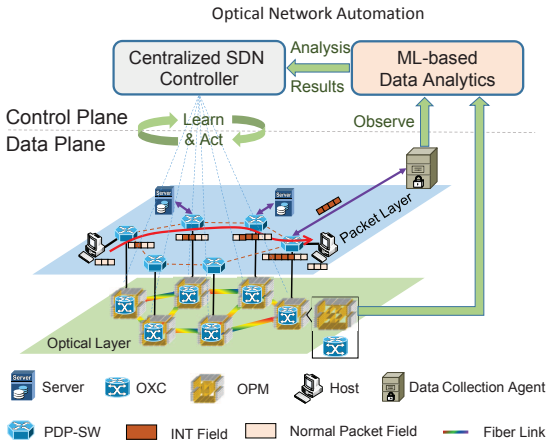


Fig. 1. Example of optical network automation in a packet-over-optical network, PDP-SW: programmable data plane switch, OXC: optical cross-connect, OPM: optical performance monitor.

module for traffic prediction (*e.g.*, the long/short-term memory based deep neural network (LSTM-DNN)) could be easily misled to make incorrect forecasts by adversarial samples whose average value was just 3.2% of the legitimate traffic samples. Later in [14], we demonstrated that the attacker's pre-knowledge about the ML-based traffic predictor could be minimized by leveraging a generative adversarial network (GAN) model and it could craft and inject adversarial samples in a more adaptive and hard-to-detect manner. However, time series prediction is just one type of tasks that ML modules do in network automation, and there are also a significant part of classification tasks for anomaly detection/location.

This motivates us to investigate how to launch adversarial-sample-based attacks to disturb the ML-based classifiers for anomaly detection in packet-over-optical networks and to analyze the consequences of the attacks in this work. Moreover, as MLaaS may require operators to submit encrypted cipher-text of telemetry data to third-party entities for ML design and training (*i.e.*, to protect the privacy of operators' networks) [19], our work considers the ML-based classifiers that were trained with plain-text and cipher-text data.

In this work, we design two adversarial-sample-based attack schemes to disturb the ML-based classifiers for anomaly detection in packet-over-optical networks. We start with assuming that the attacker can obtain information regarding the internal structure of an ML-based classifier and/or its training/testing data, and leverage the white-box attack (WBA) strategy [20] to design an attack scheme. Then, we turn to the black-box attack (BBA) strategy [20] to make the attack scheme more practical by minimizing the dependency on the pre-knowledge. For different types of ML-based classifiers that were trained with plain-text or cipher-text telemetry data, we demonstrate that our proposed BBA scheme can interact with a dynamic packet-over-optical network quietly to train itself on-the-fly, generate well-crafted adversarial samples to tamper legitimate telemetry data in the hard-to-detect way, and mislead ML-based classifiers to severely affect their performance on anomaly detection.

The rest of the paper is organized as follows. In Section II, we survey the related work. Section III describes the operation principle of the adversarial-sample-based attack schemes designed by us, and their implementation details are presented in Section IV. We explain the testbed setup for network automation in a packet-over-optical network in Section V, and discuss experimental results regarding our attack schemes in Section VI. Finally, Section VII summarizes the paper.

II. Related Work

Nowadays, the advances on flexible-grid elastic optical networks (EONs) [21, 22] have made optical infrastructures more adaptive, and the packet-over-optical networks that leverage EON have demonstrated promising advantages in today's metro/core networks, especially for those that were architected for data-center interconnects [23]. However, these technical innovations also brought in challenges on NC&M, as the network infrastructure has become more complicated and emerging services normally require multi-layer provisioning scenario [24]. This has promoted the R&D on ML-assisted network automation [6, 19, 25, 26].

The ML modules in network automation for optical or packet-over-optical networks can be roughly categorized as those for prediction and classification. As for prediction, people have leveraged various ML modules to forecast traffic volume [27] and quality-of-transmission (QoT) [28]. Classification is the most-referred-to task for ML modules in network automation systems, as it is essential to anomaly detection and fault management [6, 19]. Moreover, ML-based classification is also important for attack detection [26] and security monitoring [29]. The results of ML-based prediction and classification can be leveraged by deep reinforcement learning (DRL) based schemes for decision-making, further reducing human interventions in network automation [30–32].

However, ML modules are vulnerable to adversarial samples [13]. Our previous studies showed that ML modules for traffic prediction could be easily misled by well-crafted adversarial samples to output incorrect forecasts, severely disturbing the NC&M for virtual optical network slicing [17] and multi-layer traffic grooming and provisioning [14]. Meanwhile, the ML modules for classification are not immune to adversarial-sample-based attacks either. For instance, an attacker can easily mislead a deep neural network (DNN) for image classification with hard-to-detect adversarial samples [20].

Although how to launch adversarial-sample-based attacks to image classifiers has already been studied intensively, the approaches developed there cannot be adopted to attack the ML-based classifiers for anomaly detection. This is because the characteristics of telemetry data are fundamentally different from those of image data. Specifically, each sample of telemetry data normally contains many dimensions (*i.e.*, much larger than two), the correlation among dimensions is sparse and weak, and the value range of each dimension varies a lot [19, 26]. To the best of our knowledge, how to mislead the ML modules for anomaly detection has not been studied yet.

III. Operation Principle

In this section, we first briefly explain the architecture of the network automation system that leverages ML-based classifiers for anomaly detection in a packet-over-optical network, and

then describe the operation principle of adversarial-sample-based attacks to disturb the anomaly detection.

A. Network Automation in Packet-over-Optical Network

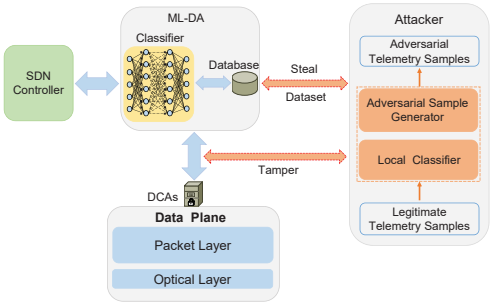
We still refer to Fig. 1 to explain the network architecture of the packet-over-optical network that utilizes network automation [6, 10]. The multi-layer data plane consists of a packet layer building over the optical layer, and both of the layers are managed by the centralized control plane. The packet layer includes programmable data plane switches (PDP-SWs) [7, 33], client hosts, and data collection agents (DCAs). Here, the DCAs are placed at the edge of the packet layer to collect the telemetry data carried by INT fields in packets. Note that, with multi-layer INT (ML-INT), telemetry data regarding both the packet and optical layers can be encoded in INT fields at each PDP-SW [10]. This is in-band network monitoring, and the control plane can also poll network elements in the data plane for their status data (*i.e.*, out-of-band network monitoring).

The optical layer is essentially an EON that is built with bandwidth-variable optical cross-connects (OXCs) and fiber links. We deploy an optical performance monitor (OPM) on each OXC to collect telemetry data (*e.g.*, power-level, bit-error-rate (BER), and optical signal-to-noise-ratio (OSNR)) regarding the passing-through lightpaths. Then, the telemetry data is sent to the PDP-SW that is local to the OXC to enable the ML-INT there. The DCAs extract and parse telemetry data in packets' INT fields to forward it to the ML-based data analytics (ML-DA) module. The ML-DA can classify the telemetry data for anomaly detection/location and report the results to the SDN controller for proper NC&M actions. Therefore, the "observe-analyze-act" decision loop can be realized to facilitate network automation.

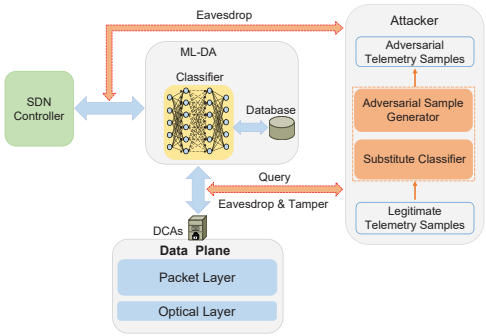
We assume that the packet-over-optical network is a metro or core network. Hence, the ML-DA and DCAs have to reside at different locations, which makes the data reporting between them vulnerable to eavesdropping and data tampering. One way to improve the security of the network automation is to let DCAs encrypt telemetry data with vector homomorphic encryption (VHE) [34] and only report the cipher-text to the ML-DA. Meanwhile, having been trained with cipher-text, the ML-DA can operate directly on encrypted telemetry data for anomaly detection/location [19]. However, this only protects the privacy of the data plane against eavesdropping, but adversarial-sample-based attacks can still be easily launched with eavesdropping and data tampering, as we will show later.

B. Principle of Adversarial-Sample-based Attacks

Apparently, the accuracy of the classifier in the ML-DA is crucial to ensure the performance of the network automation, and the simplest way to disturb the classifier's operation is to tamper its classification results directly. Moreover, it is known that ML-based classifiers are vulnerable to adversarial-sample-based attacks, *e.g.*, an image classifier can be easily misled by well-crafted adversarial samples to generate incorrect outputs [20]. As the characteristics of telemetry data are fundamentally different from those of image data, we would like to investigate



(a) White-box attack (WBA) strategy



(b) Black-box attack (BBA) strategy

Fig. 2. Operation principle of adversarial-sample-based attacks.

how badly an ML-based classifier for anomaly detection can be misled and what the consequences are.

Note that, there are three reasons for us to study the adversarial-sample-based attacks instead of assuming that the attacker tampers the classification results from the ML-DA to the SDN controller directly. Firstly, the adversarial-sample-based attacks are real security threats to network automation systems, and a comprehensive security analysis should try to cover all the potential threats. Secondly, the adversarial-sample-based attacks are more sophisticated than tampering the classification results directly, and thus they are harder to be detected. Finally, in a packet-over-optical network, there can be many DCAs and thus the amount of telemetry data being sent from the DCAs to the ML-DA can be huge [35]. Hence, when the attacker tampers the classification results directly, it might not map a classification to the corresponding telemetry data correctly. This will make its attacks easier to be detected.

Hence, in the following, we will design two adversarial-sample-based attack strategies, *i.e.*, the white-box attack (WBA) and black-box attack (BBA) strategies, based on whether the pre-knowledge about the target classifier (*i.e.*, the legitimate classifier in the ML-DA) can be obtained or not, respectively. Table I summarizes the characteristics of the two strategies. Specifically, for both of them, we assume that the attacker can tap the data reporting channels between the ML-DA and DCAs to obtain the telemetry data being transmitted in them (plain-text or cipher-text). Then, based on the telemetry data, it generates adversarial samples to tamper the legitimate samples with the man-in-the-middle attack, and makes classifiers malfunction during anomaly detection/location.

1) White-Box Attack (WBA) Strategy: Fig. 2(a) shows WBA strategy, which assumes that the attacker can steal the train-

TABLE I
SUMMARY OF ADVERSARIAL-SAMPLE-BASED ATTACKS

Strategy	Pre-knowledge	Compromised Entities
WBA	Training/Testing Sets and Architecture of Target Classifier	Data Reporting Channels between DCAs and ML-DA, and ML-DA (or MLaaS System)
BBA	None	Data Reporting Channels among DCAs, ML-DA and Controller

ing/testing sets and the architecture of the legitimate classifier. Note that, the pre-knowledge can be obtained in two ways: 1) compromising the ML-DA directly, and 2) hacking into the MLaaS system if the legitimate classifier was outsourced to a third party [16]. Hence, WBA can be launched if either of these two ways is feasible, which is possible because there are a few techniques [36] for the attacker to hack into the database of the ML-DA or the MLaaS system. Then, by analyzing the information, the attacker can choose a proper ML model to design a substitute classifier, and train it to imitate the operation of the legitimate classifier in ML-DA. Next, the trained substitute classifier can craft and inject adversarial telemetry samples adaptively to disturb the legitimate classifier.

2) Black-Box Attack (BBA) Strategy: The attacker can access the training/testing sets and the architecture of the legitimate classifier is actually a relatively strong assumption. Therefore, we design the BBA strategy in Fig. 2(b) for the cases in which such pre-knowledge is not available. By eavesdropping the data reporting channels between the ML-DA and DCAs, the attacker can collect a set of legitimate telemetry samples, and use them to generate training samples for a substitute classifier. Specifically, it sends artificial telemetry samples to the legitimate classifier through a data reporting channel, observes the ML-DA's output for data labeling by tapping the connection between the SDN controller and ML-DA, and gets training samples[1]. Then, it trains a substitute classifier with the training samples. The attacker crafts adversarial examples with the substitute classifier, and injects them back to the data reporting channels for misleading the legitimate classifier in the ML-DA.

IV. DESIGN OF BBA STRATEGY

As WBA becomes infeasible when pre-knowledge about the legitimate classifier cannot be obtained, we focus on BBA to design our adversarial-sample-based attack scheme.

A. Threat Model

We define the input to an ML-based classifier as a multi-dimensional vector $\vec{x}$, each dimension of which represents a type of telemetry data regarding the packet-over-optical network (*e.g.*, packet processing latency or OSNR). The ML-based classifier can be denoted as $\mathcal{T}$, which is the target of BBA. As we consider a BBA strategy, we assume that the attacker does not have any access to the private information of $\mathcal{T}$, such as internal structure and training/testing sets, while it should be able to query $\mathcal{T}$ by sending in an arbitrary input $\vec{x}$ and collecting the classified label $\mathcal{T}(\vec{x})$. Then, the objective of the attacker is to produce a minimally-altered version of $\vec{x}$, *i.e.*, the adversarial sample $\vec{x^*}$, which can mislead $\mathcal{T}$ to output $\mathcal{T}(\vec{x^*}) \neq \mathcal{T}(\vec{x})$. The attack is on the classifier's output integrity [36], and the adversarial sample is

$$\vec{x^*} = \vec{x} + \text{argmin}\left[\vec{\vartheta} : \mathcal{T}(\vec{x} + \vec{\vartheta}) \neq \mathcal{T}(\vec{x})\right] = \vec{x} + \delta_{\vec{x}}, \quad (1)$$

where $\delta_{\vec{x}}$ is the minimal perturbation to realize the attack. Note that, solving the optimization to obtain $\delta_{\vec{x}}$ can be challenging, because the characteristics of the ML-based classifier can make the optimization neither linear nor convex, especially when the internal structure of the classifier is unknown.

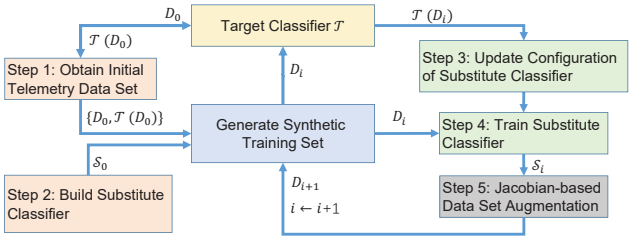


Fig. 3. Training of substitute classifier.

B. Procedure of BBA

In order to launch attacks with the BBA strategy, the attacker needs to train a substitute classifier locally, for imitating the operation of the ML-based classifier in the ML-DA (as shown in Fig. 2(b)). Therefore, by eavesdropping data reporting channels to obtain a legitimate telemetry sample $\vec{x}$ and inputting it to the substitute classifier, the attacker can obtain the minimal perturbation $\delta_{\vec{x}}$ to craft the adversarial sample $\vec{x^*}$. However, the fundamental difficulty of such a strategy is how to obtain a proper training set to train the substitute classifier efficiently, as the telemetry samples reported from DCAs to the ML-DA might only contain sparse information for the training of the substitute classifier (*i.e.*, the packet-over-optical network operates in its normal state in most of the time [37]).

Hence, in the following, we will design the procedure of B-BA, which enables the attacker to generate a synthetic training set to effectively improve its training efficiency. Specifically, we propose a scheme to enable the attacker to train the substitute classifier with a synthetic training set, which is generated by leveraging a small set of legitimate telemetry samples. By observing the synthetic set, the attacker selects a proper architecture $\mathcal{S}$ to build its substitute classifier for imitating $\mathcal{T}$. In other words, the procedure of our proposed BBA scheme includes two steps: 1) training of the substitute classifier $\mathcal{S}$ and 2) crafting of adversarial samples.

1) Training of Substitute Classifier: As shown in Fig. 3, we realize the training of the substitute classifier as follows.

[1]Note that, encrypting the communications between the SDN controller and the ML-DA can make it more difficult for the attacker to launch adversarial-sample-based attacks. However, the encryption will bring in additional operational complexity and cost, especially to increase the computation loads on both parties and prolong their response time. As the real-time performance of the SDN controller and the ML-DA is essential to the timeliness of network automation, we, in this work, assume that the ML-DA sends the classification results (*i.e.*, labels of anomaly types) to the SDN controller in plain-text [19].

- **Step 1**: The attacker hacks into the data reporting channels between the ML-DA and DCAs and eavesdrops a small set of legitimate telemetry samples D_0. It then queries the ML-based classifier $\mathcal{T}$ (*i.e.*, the target classifier) to label all the telemetry samples in D_0 as $\mathcal{T}(D_0)$, and obtains the initial training set $\{D_0, \mathcal{T}(D_0)\}$.
- **Step 2**: The attacker builds the substitute classifier as $\mathcal{S}_0$, for imitating the target classifier $\mathcal{T}$ in the ML-DA.
- **Step 3**: The attacker updates the configuration of the substitute classifier $\mathcal{S}_i$ (*i.e.*, $i = 0$ initially) according to the number of anomaly types in the current training set ($\{D_i, \mathcal{T}(D_i)\}$). Specifically, each known anomaly type should correspond to an output in the output layer of $\mathcal{S}_i$.
- **Step 4**: The substitute classifier $\mathcal{S}_i$ is trained with the back-propagation and gradient-descent algorithm [38].
- **Step 5**: With the trained substitute classifier $\mathcal{S}_i$, the attacker applies the augmentation technique on D_i to generate a synthetic telemetry sample set D'_i, queries the target classifier $\mathcal{T}$ to get a labeled data set $\{D'_i, \mathcal{T}(D'_i)\}$, merges it with $\{D_i, \mathcal{T}(D_i)\}$ to get a larger training set $\{D_{i+1}, \mathcal{T}(D_{i+1})\}$, and increases i to $i + 1$.

Then, by repeating **Steps 3-5**, the attacker can get a substitute classifier $\mathcal{S}_{i_{\max}}$, which has been trained with a reasonably large synthetic training set $\{D_{i_{\max}}, \mathcal{T}(D_{i_{\max}})\}$, and the accuracy of $\mathcal{S}_{i_{\max}}$ for imitating the target classifier $\mathcal{T}$ can be controlled by selecting a proper maximum iteration number $i_{\max}$.

Algorithm 1 explains how to train the substitute classifier. *Line* 1 is for the initialization, where $\{\tau, \gamma, \kappa\}$ are preset coefficients for controlling the procedure of the training, and other training parameters (*e.g.*, the maximum number of training epochs) are also initialized here. Then, the initial training set $\{D_0, \mathcal{T}(D_0)\}$ is obtained in *Line* 2. The for-loop covering *Lines* 3-16 conducts the augmentation of current training set and the training of substitute classifier $\mathcal{S}_i$ in $i_{\max}$ iterations. We first build ($i = 0$) or update ($i > 0$) the substitute classifier $\mathcal{S}_i$ according to $|\mathcal{T}(D_i)|$, which refers to the number of anomaly types in the current training set $\{D_i, \mathcal{T}(D_i)\}$ (*Line* 4). Specifically, the number of neurons in the output layer of $\mathcal{S}_i$ should equal $|\mathcal{T}(D_i)|$. Then, *Line* 5 trains the substitute classifier $\mathcal{S}_i$ with the current training set $\{D_i, \mathcal{T}(D_i)\}$.

The augmentation technique to expand the current training set is applied in *Lines* 6-15. We first introduce the procedure in *Lines* 6-10 to reduce the complexity of the augmentation. Specifically, if the number of iterations i reaches the preset threshold γ, we random select κ samples from $\{D_i, \mathcal{T}(D_i)\}$ to form the data set $\{D_s, \mathcal{T}(D_s)\}$ for augmentation (*Lines* 6-7), *i.e.*, the size of $\{D_s, \mathcal{T}(D_s)\}$ will be smaller than that of $\{D_i, \mathcal{T}(D_i)\}$. Otherwise, the whole $\{D_i, \mathcal{T}(D_i)\}$ will be used for augmentation (*Lines* 8-9). The augmentation is based on identifying the direction in which the output of the substitute classifier $\mathcal{S}_i$ is changing during its training, which is achieved by checking the sign of its Jacobian matrix $J_{\mathcal{S}_i}$ (*Line* 12).

Specifically, we denote the sign for an input sample $\vec{x}$ as $sign\left(J_{\mathcal{S}_i}[\mathcal{T}(\vec{x})]\right)$, and the new synthetic data set is

$$D'_i = \{\vec{x} + \lambda_i \cdot sign(J_{\mathcal{S}_i}[\mathcal{T}(\vec{x})]) : \vec{x} \in D_s\}, \tag{2}$$

where the step-size λ_i alternates between positive and negative values to improve the approximation made by the substitute classifier. *Line* 11 explains how to update the step-size, where τ is set to be the number of iterations after which the augmentation does not lead to any substantial improvement on the approximation. After obtaining the new synthetic data set D'_i (*Line* 13), we label it by querying the target classifier $\mathcal{T}$ and merge the results with the current training set to obtain an enhanced training set $\{D_{i+1}, \mathcal{T}(D_{i+1})\}$ (*Lines* 14-15). Finally, the substitute model used to craft adversarial samples (*i.e.*, $\mathcal{S}_{i_{\max}}$) is obtained in *Lines* 17-18.

Algorithm 1: Training of Substitute Classifier

Input: Target classifier $\mathcal{T}$, maximum iteration number $i_{\max}$, initial telemetry data set D_0, initial step-size of Jacobian-based data set augmentation λ_0.

Output: Parameters $\theta_{\mathcal{S}_{i_{\max}}}$ of substitute classifier $\mathcal{S}_{i_{\max}}$.

1 initialize coefficients $\{\tau, \gamma, \kappa\}$ and parameters of training;
2 query the target classifier $\mathcal{T}$ to label samples in D_0 and obtain the initial training set $\{D_0, \mathcal{T}(D_0)\}$;
3 **for** $i \in [0, i_{\max} - 1]$ **do**
4 build/update substitute classifier $\mathcal{S}_i$ based on $|\mathcal{T}(D_i)|$;
5 train $\mathcal{S}_i$ with current training set $\{D_i, \mathcal{T}(D_i)\}$ to get its parameters $\theta_{\mathcal{S}_i}$;
6 **if** $i > \gamma$ **then**
7 select κ samples randomly from current training set $\{D_i, \mathcal{T}(D_i)\}$ to form a set $\{D_s, \mathcal{T}(D_s)\}$;
8 **else**
9 $\{D_s, \mathcal{T}(D_s)\} = \{D_i, \mathcal{T}(D_i)\}$;
10 **end**
11 compute the step-size $\lambda_i = \lambda_0 \cdot (-1)^{\lfloor \frac{i}{\tau} \rfloor}$;
12 compute the sign of Jacobian matrix $J_{\mathcal{S}_i}[\mathcal{T}(D_s)]$;
13 perform Jacobian-based data set augmentation with Eq. (2) to obtain D'_i;
14 query $\mathcal{T}$ to label samples in D'_i as $\{\mathcal{T}(\vec{x}),\ \vec{x} \in D'_i\}$;
15 $D_{i+1} = D_i \cup D'_i$, $\mathcal{T}_{i+1} = \mathcal{T}_i \cup \mathcal{T}'_i$, $i = i + 1$;
16 **end**
17 update configuration of substitute classifier $\mathcal{S}_{i_{\max}}$ based on $|\mathcal{T}(D_{i_{\max}})|$;
18 train $\mathcal{S}_{i_{\max}}$ with current training set $\{D_{i_{\max}}, \mathcal{T}(D_{i_{\max}})\}$ to obtain its parameters $\theta_{\mathcal{S}_{i_{\max}}}$;
19 **return**($\theta_{\mathcal{S}_{i_{\max}}}$);

2) Crafting of Adversarial Samples: With the trained substitute classifier, the attacker can leverage the approaches developed in [39, 40] to craft adversarial samples, because they can scale with large telemetry data sets time-efficiently. Specifically, the DeepFool in [39] and the FGSM in [40] share the similar principle of evaluating the substitute classifier's sensitivity to input modifications to find the perturbation for achieving the goal of misclassification. Each of them has pros and cons when being applied to our problem. FGSM can generate many adversarial samples quickly with larger perturbations, while DeepFool can find the smallest perturbations at the expense of a larger computational complexity.

Intuitively, an ML-based classifier can be misled more easily and more quietly if the adversarial samples are closer to its decision boundaries. To generate such adversarial samples, DeepFool finds the minimal perturbations based on the dis-

tance between a sample data point and the boundary of the data's hyperplane. Specifically, for a target classifier $\mathcal{T}$, it defines an adversarial perturbation as the minimal perturbation $\delta_{\vec{x}}$ that is sufficient to change the labeling output $\mathcal{T}(\vec{x})$

$$\Delta(\vec{x};\mathcal{T}) := \min(||\delta_{\vec{x}}||_\infty),\ \mathcal{T}(\vec{x}+\delta_{\vec{x}}) \neq \mathcal{T}(\vec{x}), \tag{3}$$

FGSM associates a cost function $\mathcal{C}(\cdot)$ with $\mathcal{T}$, for crafting an adversarial sample $\vec{x}^*$ based on a legitimate sample $\vec{x}$

$$\vec{x^*} = \vec{x} + \epsilon \cdot sign\left(\nabla_{\vec{x}}\mathcal{C}(\mathcal{T}(\vec{x}),\vec{x})\right), \tag{4}$$

where the cost function evaluates the perturbation $\delta_{\vec{x}}$, $sign\left(\nabla_{\vec{x}}\mathcal{C}(\mathcal{T}(\vec{x}),\vec{x})\right)$ denotes the sign of the gradient of the cost function with respect to $\vec{x}$, and the parameter ϵ controls the amplitude of the perturbation. Both the probability of $\vec{x}$ being misclassified by $\mathcal{T}$ and the likelihood of the adversarial attack being detected increase with ϵ. Therefore, we can adjust the value of ϵ to balance the tradeoff.

V. Experiments for Data Collection

This section explains how we set up our experimental testbed of a packet-over-optical network and conduct experiments in it to collect telemetry data for anomaly detection.

A. Testbed Setup

We set up a small but real packet-over-optical network testbed as the multi-layer data plane for anomaly detection with ML-based classifiers. The optical layer is built with bandwidth-variable wavelength-selective switches (BV-WSS') and an optical line system (OLS). The BV-WSS' are commercial products, each of which has a configuration of 1×9 and provides a spectrum allocation granularity of 12.5 GHz to enable flexible-grid lightpath provisioning [41]. The OLS is based on the Juniper BTI7800 platform, which deploys bandwidth-variable transponders (BV-Ts) on nodes and in-line erbium-doped fiber amplifiers (EDFAs) on fiber links, for establishing lightpaths. Each BV-T can support line-rates within $[100, 400]$ Gbps. We also insert an OPM on each node in the optical layer to collect telemetry data regarding active lightpaths (*e.g.*, power-level, OSNR, and optical spectrum).

The packet layer consists of client hosts, PDP-SWs and DCAs. We realize each PDP-SW based on the high-performance software switch developed in [35], which runs on a Linux server and is capable of ML-INT. Note that, in addition to telemetry data regarding the optical layer, the ML-INT can also collect that about the packet layer, *e.g.*, bandwidth usage, packet forwarding behavior, and packet processing latency. Each host is emulated with a commercial traffic generator/analyzer that can generate/receive application traffic at 10/40 Gbps. The DCAs are homemade and run on high-performance Linux servers to collect and process telemetry data in real-time (with a throughput of 2 million packets per second (Mpps) per port).

The control plane system is realized by extending the open network operating system (ONOS) platform [42], which also runs on a high-performance Linux server. In this work, we assume that telemetry data can be reported to the control plane in either the in-band or out-of-band manner. In the in-band manner, the telemetry data is inserted in packets as INT fields by PDP-SWs and collected at network edge by the DCAs [10], which is for real-time and fine-grained network monitoring. The out-of-band manner lets OPMs report telemetry data directly to the control plane to respond to polling requests, which is for coarse-grained network monitoring.

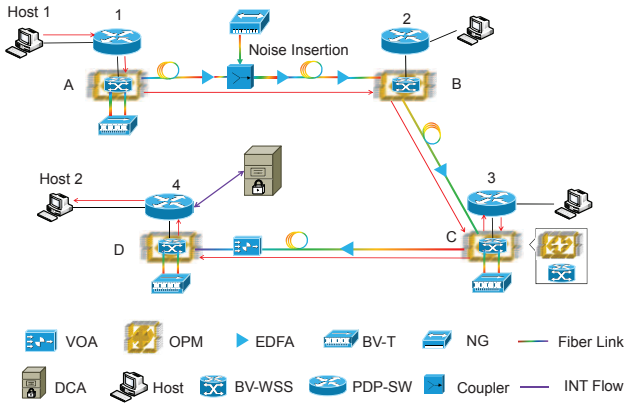


Fig. 4. Experimental setup for anomaly detection in a multi-layer packet-over-optical network, VOA: Variable optical attenuator, BV-T: Bandwidth-variable transponder, EDFA: Erbium-doped fiber amplifier, BV-WSS: Bandwidth-variable wavelength-selective switch, NG: Noise generator.

B. Data Collection

With the testbed, we conduct experiments for telemetry data collection and multi-layer anomaly detection. As shown in Fig. 4, we make the host connecting to *PDP-SW* 1 transmit a packet flow to the one connecting to *PDP-SW* 4. The flow is routed in the packet-over-optical network as indicated by the red solid line in Fig. 4, *i.e.*, *Host* 1 $\rightarrow$ *PDP-SW* 1 $\rightarrow$ *BV-WSS* A $\rightarrow$ *BV-WSS* B $\rightarrow$ *BV-WSS* C $\rightarrow$ *PDP-SW* 3 $\rightarrow$ *BV-WSS* C $\rightarrow$ *BV-WSS* D $\rightarrow$ *PDP-SW* 4 $\rightarrow$ *Host* 2. Hence, the multi-layer routing of the flow involves three lightpaths in the optical layer and three PDP-SWs in the packet layer.

With the testbed and flow configuration in Fig. 4, we apply various network settings, including EDFA settings for noise insertion, BV-WSS settings for filter offset, spectrum assignments for the lightpaths, bandwidth usages in the packet layer, and flow-table configurations in the PDP-SWs, to emulate both normal and abnormal cases in the packet-over-optical network, and collect $\sim95,000$ telemetry data samples. Each sample includes six elements, *i.e.*, OSNR, input power, chromatic dispersion (CD), packet forwarding latency, bandwidth usage on input port (Input BW), and bandwidth usage on output port (Output BW), and thus each sample is 6-dimensional. Then, according to the actual root-cause of its anomaly (if there is any), we label each sample with "Normal", "High Power", "Low Power", "Degraded OSNR", "WSS Left Shift", "WSS Right Shift", "High Delay", "Packet Congestion", "Packet Loss", and "Switch Misconfiguration". We put 90% and 10% of the samples in training and testings sets, respectively, and train the legitimate ML-based classifier accordingly.

We also encrypt the samples with VHE [34] and train the ML-based classifier with cipher-text to cover the case with MLaaS. Therefore, for each type of ML-based classifier, we actually train two classifiers with plain-text and cipher-text

data sets, respectively. For instance, if we leverage the DNN structure in [19], the training of the classifiers for plain-text and cipher-text data sets can be finished in $4,666.37$ and $4,658.73$ seconds, respectively, and they can respectively achieve classification accuracies of 99.99% and 99.64% on their testing sets. In order to evaluate the performance of adversarial-sample-based attacks, we introduce two metrics. The **success rate** is the proportion of adversarial samples that can make the substitute classifier categorize them into incorrect anomaly types. The **transferability rate** of adversarial samples refers to the misclassification rate of the legitimate classifier on adversarial samples crafted by the attacker.

VI. Experiments of Adversarial-Sample-based Attacks

In this section, we conduct experiments to validate the threats of adversarial-sample-based attacks on ML-based classifiers for network automation. We first assume that the classifier uses the DNN structure in [19] and performs WBA and BBA on it, and then generalize the results by considering more structures for the ML-based classifier.

A. Performance of WBA

As we have explained in Section III-B, the adversarial-sample-based attack with WBA is relatively straightforward, because the attacker can access both the internal structure and training/testing data sets of the legitimate classifier. Therefore, the attacker can use the same structure to design the local classifier and train it accordingly. In this case, the success rate and transferability rate become identical. We craft adversarial samples with DeepFool and FGSM to implement the WBA. Fig. 5 shows the success rate of the attacks, where the "perturbation degree" on the x-axis refers to Δ in Eq. (3) and ϵ in Eq. (4) for DeepFool and FGSM, respectively. To ensure sufficient statistical accuracy, all of our simulations average the results from 10 independent runs to obtain each data point.

The results show that in all the scenarios, the ML-based classifiers can be misled by the adversarial-sample-based attacks, and the smallest success rate is above 30% even when the perturbation degree is only 0.075. The success rate first increases rapidly with the perturbation degree and then converges. Between DeepFool and FGSM, FGSM can achieve better adversarial-sample-based attacks as it provides a higher success rate when other parameters are the same. We also notice that when the same adversarial-sample-crafting algorithm is used, the classifier trained with cipher-text is always more vulnerable than that trained with plain-text. This suggests that even though VHE can protect the privacy of operators better in MLaaS, it also brings in new vulnerability when adversarial-sample-based attacks can be launched.

B. Performance of BBA

Next, we consider the BBA strategy and assume that the initial legitimate data set only contains 570 samples (*i.e.*, $|D_0| = 570$), in which there should be at least one sample for each type of anomaly. The attacker architects the substitute classifier with a linear DNN model for classification, which consists of 7 layers. The coefficients of *Algorithm* 1 are set as $\tau = 5$, $\gamma = 5$, and $\kappa = 400$, and *Algorithm* 1 trains the substitute classifier with $i_{\max} = 10$ and $\lambda_0 = 0.1$. This leads to classification accuracies of 95.75% and 89.44% on plain-text and cipher-text data sets, respectively. With the trained substitute classifiers, the attacker crafts adversarial samples with DeepFool and FGSM, as it does in WBA.

Fig. 6 indicates that FGSM still crafts adversarial samples better than DeepFool. Specifically, when other parameters are the same, both the success rate and transferability rate achieved by FGSM are higher than those from DeepFool. As for FGSM with the perturbation degree of $\epsilon = 0.15$, the transferability rates of the substitute classifiers on plain-text and cipher-text data sets are 88.86% and 86.83%, respectively. Hence, the adversarial-sample-based attacks can indeed damage the output integrity of the legitimate ML-based classifiers severely.

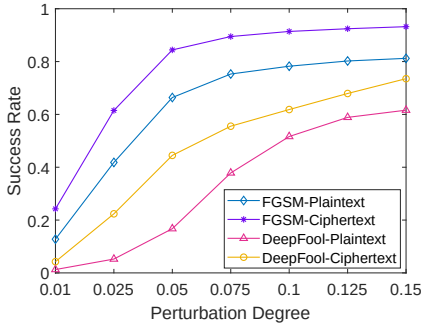
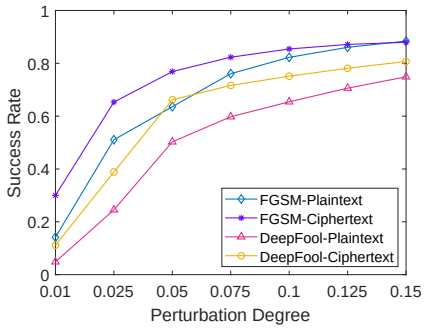
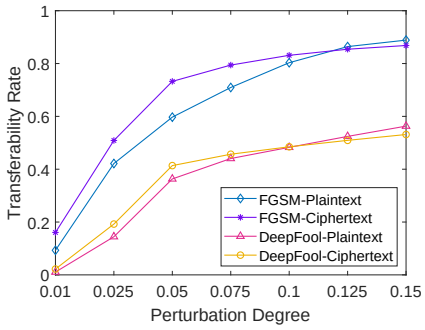


Fig. 5. Success rate of adversarial-sample-based attacks with WBA.

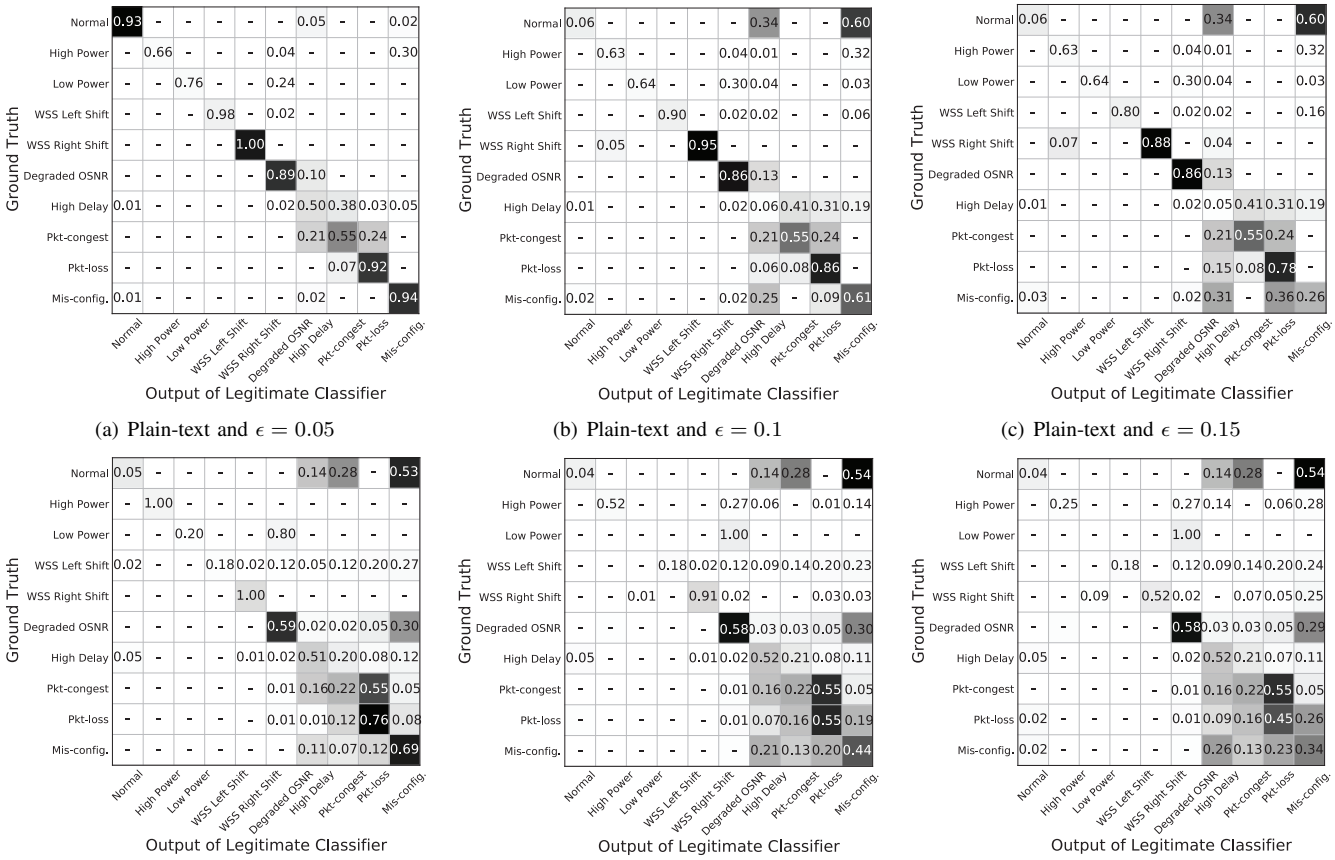


(a) Success Rate



(b) Transferability Rate

Fig. 6. Results of adversarial-sample-based attacks in BBA.



(a) Plain-text and $\epsilon = 0.05$ (b) Plain-text and $\epsilon = 0.1$ (c) Plain-text and $\epsilon = 0.15$

(d) Cipher-text and $\epsilon = 0.05$ (e) Cipher-text and $\epsilon = 0.1$ (f) Cipher-text and $\epsilon = 0.15$

Fig. 7. Confusion matrices caused by the adversarial-sample-based attacks with DeepFool.

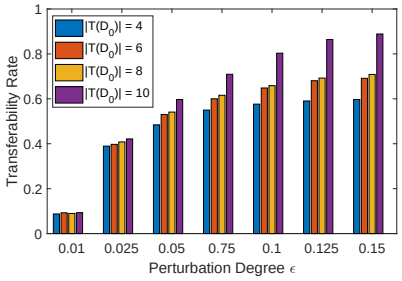
Meanwhile, with the same adversarial-sample-crafting algorithm, the legitimate classifier trained with cipher-text is still more vulnerable than that trained with plain-text.

Finally, we plot the confusion matrices of the adversarial-sample-based attack with DeepFool in Fig. 7 to further analyze the effects of the attacks. Here, we only show the results of DeepFool because the confusion matrices of DeepFool and FGSM are similar and DeepFool actually performs slightly worse. In other words, we would like to show the lower-bound of our proposals' performance. Fig. 7 illustrates that the attacks make the target classifier generate random classification errors, which are difficult to detect and protect against. Meanwhile, the results indicate that more telemetry data samples will be misclassified as "High Delay", "Packet Loss", and "Switch Misconfiguration" as ϵ increases. This is because they are the most common anomaly types in network operation. Moreover, we can see that there are more misclassifications in the cases with cipher-text when the perturbation degree is the same. This is because VHE encrypts one sample to different cipher-text ones in different rounds [19], which is similar as adding a small random perturbation to each cipher-text sample.

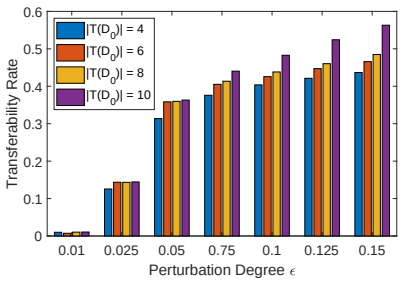
C. Performance of BBA with Incomplete Anomaly Types

To further verify the practicalness of our proposals, we design simulations to consider the cases in which *Algorithm* 1 has to start with an initial legitimate telemetry data set D_0 that only contains an incomplete list of anomaly types. Note that, when the D_0 eavesdropped by the attacker does not contain all the anomaly types, it can learn more anomaly types in two ways: 1) conducting additional eavesdropping to collect more legitimate telemetry data samples, and 2) applying the augmentation technique in *Algorithm* 1 on its current data set to generate more synthetic telemetry data samples. Although both ways are feasible and they can be used simultaneously, we design our simulations to only consider the latter one, for the reason that eavesdropping activities should be minimized to reduce the probability of being detected. Therefore, since the restriction applied to the attacker is stricter than that in practical scenarios, the results in this subsection will show the lower-bound of the performance of *Algorithm* 1.

Specifically, the simulations consider four cases, wherein the number of anomaly types in the initial legitimate telemetry data set D_0 is set as $|\mathcal{T}(D_0)| = \{4, 6, 8, 10\}$ (*i.e.*, D_0 contains 390, 450, 510 and 570 samples, respectively). Except for these, the overall structure of the substitute classifier and the parameters used in *Algorithm* 1 remain unchanged. Then, we respectively obtain the classification accuracy of the substitute classifier as 67.21%, 77.27%, 79.81%, and 95.75% on plain-text data sets, and as 72.25%, 78.55%, 81.62% and 89.44% on cipher-text data sets. As expected, the classification accuracy of the substitute classifier increases with $|\mathcal{T}(D_0)|$, *i.e.*, *Algorithm* 1 can train a better substitute classifier when the initial

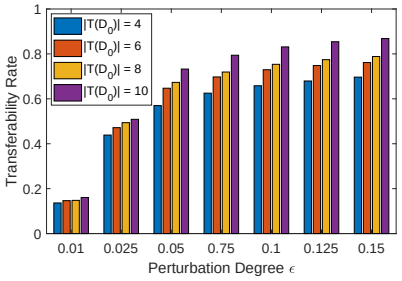


(a) FGSM

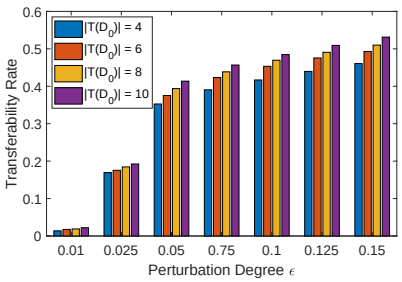


(b) DeepFool

Fig. 8. Transferability rate of adversarial-sample-based attacks (plain-text).



(a) FGSM



(b) DeepFool

Fig. 9. Transferability rate of adversarial-sample-based attacks (cipher-text).

legitimate telemetry data set provides more information about the anomaly types. Note that, when we have $|\mathcal{T}(D_0)| = 10$, D_0 contains the complete list of anomaly types and thus the classification accuracies are the same as those in Section VI-B.

Then, we let the attacker craft adversarial samples with DeepFool and FGSM based on the trained substitute classifiers. Figs. 8 and 9 show the results on transferability rate. We observe that the transferability rates of the adversarial-sample-based attacks, which start to craft the adversarial samples with different initial legitimate data sets, are comparable in all the cases. This confirms the effectiveness and practicalness of *Algorithm* 1, *i.e.*, it can build and train a reasonably good substitute classifier even when D_0 only contains an incomplete list of anomaly types. Specifically, with the perturbation degree of $\epsilon = 0.05$, the transferability rates of FGSM and DeepFool are higher than 40% and 30%, respectively, on both plain-text and cipher-text data sets, even when we have $|\mathcal{T}(D_0)| = 4$. This suggests that *Algorithm* 1 can generate synthetic data samples that cover all the anomaly types, even when it is given an initial legitimate data set that only contains 40% of the anomaly types. Hence, the substitute classifier can be trained to effectively imitate the operation of the target classifier.

We also observe that the transferability rate increases with $|\mathcal{T}(D_0)|$, and it increases more significantly when plain-text is considered. This further confirms that cipher-text is more vulnerable to data tampering. FGSM still crafts adversarial samples better than DeepFool when other things are the same.

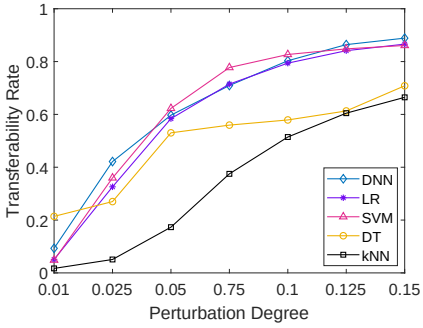
D. Generalization of BBA

In order to verify that our BBA scheme can affect different types of ML-based classifiers for anomaly detection in general, we leverage logistic regression (LR), support vector machine (SVM), decision tree (DT), and k nearest neighbors (kNN) to architect the legitimate classifier and apply adversarial-sample-based attacks to them. Here, the substitute classifier used by the attacker is still based on the same DNN architecture and gets trained in the same way as in previous subsections. We respectively denote the legitimate and substitute classifiers as $\mathcal{T}$ and $\mathcal{S}$ and list their accuracies on the testing data sets in Table II. It can be seen that the substitute classifiers trained with *Algorithm* 1 still achieve reasonably good classification accuracies, even though their structures are different from those of the legitimate ones. As the classification accuracies of the substitute classifiers are the lowest when the legitimate classifiers are designed with DT and kNN, they show more robustness against the adversarial-sample-based attacks.

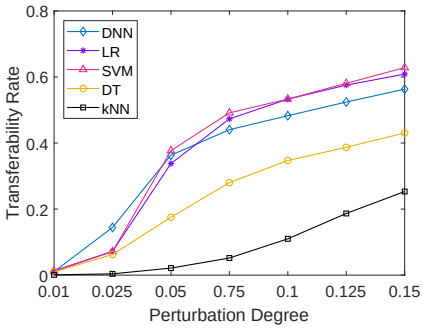
TABLE II
PERFORMANCE OF ML-BASED ANOMALY DETECTION

Accuracy on Plain-text Data Set					
	Legitimate Classifier				
	DNN	LR	SVM	DT	kNN
$\mathcal{T}$	99.99%	96.96%	96.51%	92.83%	99.64%
$\mathcal{S}$ (DNN)	95.75%	95.34%	95.46%	85.31%	82.13%
Accuracy on Cipher-text Data Set					
	Legitimate Classifier				
	DNN	LR	SVM	DT	kNN
$\mathcal{T}$	99.64%	92.08%	91.42%	89.39%	97.14%
$\mathcal{S}$ (DNN)	89.44%	91.18%	90.80%	77.84%	76.08%

We then apply DeepFool and FGSM to craft adversarial samples, and the results on transferability rate are shown in Figs. 10 and 11, for the cases with plain-text and cipher-text data sets, respectively. We observe that all the legitimate classifiers are still vulnerable to the adversarial samples crafted with FGSM and DeepFool, which indicates that the absence of

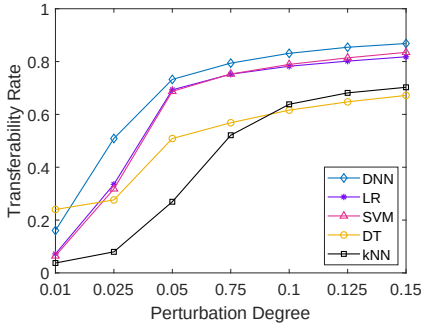


(a) FGSM

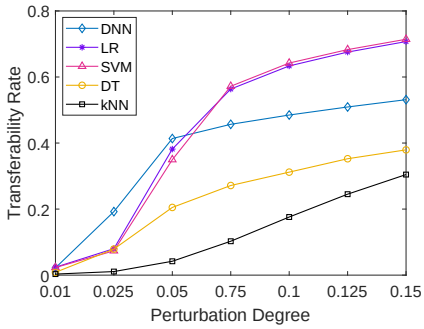


(b) DeepFool

Fig. 10. Transferability rate of adversarial-sample-based attacks (plain-text).



(a) FGSM



(b) DeepFool

Fig. 11. Transferability rate of adversarial-sample-based attacks (cipher-text).

pre-knowledge on the structure of the legitimate classifier does not make it more difficult to launch adversarial-sample-based attacks. Meanwhile, it can be seen that the cases with cipher-text data sets are still more vulnerable to data tampering even when the legitimate classifier can use different structures. The transferability rates of the cases when the legitimate classifiers are designed with DT and kNN are generally lower than those of other cases, which suggests that they are the two most robust architectures to protect against the adversarial-sample-based attacks. However, whether they are more suitable for the legitimate classifiers in practical network automation systems needs further investigations. This is because the classification accuracy of DT is normally lower than other types of classifiers and the operation of kNN can be time-consuming.

VII. Conclusion

This paper conducted an adversarial analysis of the ML-based anomaly detection in packet-over-optical networks. To check how secure the anomaly detection was, we designed two adversarial-sample-based attack schemes respectively based on WBA and BBA strategies, which eavesdropped and tampered legitimate telemetry samples to generate adversarial samples adaptively, for disturbing ML-based classifiers and in turn misleading the network automation system of a packet-over-optical network to make incorrect NC&M decisions. With a real-world testbed, we demonstrated that our proposed schemes could monitor and interact with a dynamic packet-over-optical network to train itself, such that adversarial samples could be generated and injected in the network automation system in the hard-to-detect way. The results showed that our schemes could mislead different types of ML-based classifiers to severely affect their performance on anomaly detection.

Acknowledgments

This work was supported by NSFC project 61871357 and Fundamental Fund for Central Universities (WK3500000006).

References

[1] P. Lu *et al.*, "Highly-efficient data migration and backup for Big Data applications in elastic optical inter-datacenter networks," *IEEE Netw.*, vol. 29, pp. 36–42, Sept./Oct. 2015.

[2] L. Gong and Z. Zhu, "Virtual optical network embedding (VONE) over elastic optical networks," *J. Lightw. Technol.*, vol. 32, pp. 450–460, Feb. 2014.

[3] L. Gong, Y. Wen, Z. Zhu, and T. Lee, "Toward profit-seeking virtual network embedding algorithm via global resource capacity," in *Proc. of INFOCOM 2014*, pp. 1–9, Apr. 2014.

[4] M. Zeng, W. Fang, and Z. Zhu, "Orchestrating tree-type VNF forwarding graphs in inter-DC elastic optical networks," *J. Lightw. Technol.*, vol. 34, pp. 3330–3341, Jul. 2016.

[5] J. Liu *et al.*, "On dynamic service function chain deployment and readjustment," *IEEE Trans. Netw. Serv. Manag.*, vol. 14, pp. 543–553, Sept. 2017.

[6] S. Tang, J. Kong, B. Niu, and Z. Zhu, "Programmable multilayer INT: An enabler for AI-assisted network automation," *IEEE Commun. Mag.*, vol. 58, pp. 26–32, Jan. 2020.

[7] S. Li *et al.*, "Protocol oblivious forwarding (POF): Software-defined networking with enhanced programmability," *IEEE Netw.*, vol. 31, pp. 12–20, Mar./Apr. 2017.

[8] D. Rafique and L. Velasco, "Machine learning for network automation: overview, architecture, and applications," *J. Opt. Commun. Netw.*, vol. 10, pp. D126–D143, Oct. 2018.

[9] C. Kim *et al.*, "In-band network telemetry (INT)," *Tech. Spec.*, Jun. 2016. [Online]. Available: https://p4.org/assets/INT-current-spec.pdf

[10] B. Niu *et al.*, "Visualize your IP-over-optical network in realtime: A P4-based flexible multilayer in-band network telemetry (ML-INT) system," *IEEE Access*, vol. 7, pp. 82 413–82 423, Aug. 2019.

[11] S. Liu *et al.*, "Highly-efficient and automatic spectrum inspection based on AutoEncoder and semi-supervised learning for anomaly detection in EONs," *J. Lightw. Technol.*, vol. 39, pp. 1243–1254, Mar. 2021.

[12] W. Lu *et al.*, "AI-assisted knowledge-defined network orchestration for energy-efficient data center networks," *IEEE Commun. Mag.*, vol. 58, pp. 86–92, Jan. 2020.

[13] N. Papernot *et al.*, "The limitations of deep learning in adversarial settings," in *Proc. of Euro S&P 2016*, pp. 372–387, Mar. 2016.

[14] M. Wang, H. Lu, S. Liu, and Z. Zhu, "How to mislead AI-assisted network automation in SD-IPoEONs: A comparison study of DRL- and GAN-based approaches," *J. Lightw. Technol.*, vol. 38, pp. 5574–5585, Oct. 2020.

[15] M. Brinkmann *et al.*, "ALPACA: Application layer protocol confusion-analyzing and mitigating cracks in TLS authentication," in *Proc. of SSYM 2021*, pp. 4293–4310, Aug. 2021.

[16] M. Ribeiro, K. Grolinger, and M. Capretz, "MLaaS: Machine learning as a service," in *Proc. of ICMLA 2015*, pp. 896–902, Dec. 2015.

[17] J. Guo and Z. Zhu, "When deep learning meets inter-datacenter optical network management: Advantages and vulnerabilities," *J. Lightw. Technol.*, vol. 36, pp. 4761–4773, Oct. 2018.

[18] M. Wang, S. Liu, and Z. Zhu, "Can you trust AI-assisted network automation? a DRL-based approach to mislead the automation in SD-IPoEONs," in *Proc. of OFC 2020*, pp. 1–3, Mar. 2020.

[19] X. Pan *et al.*, "Privacy-preserving multilayer in-band network telemetry and data analytics: For safety, please do not report plaintext data," *J. Lightw. Technol.*, vol. 38, pp. 5855–5866, Nov. 2020.

[20] N. Akhtar, A. Mian, N. Kardan, and M. Shah, "Advances in adversarial attacks and defenses in computer vision: A survey," *IEEE Access*, vol. 9, pp. 155 161–155 196, Nov. 2021.

[21] Z. Zhu, W. Lu, L. Zhang, and N. Ansari, "Dynamic service provisioning in elastic optical networks with hybrid single-/multi-path routing," *J. Lightw. Technol.*, vol. 31, pp. 15–22, Jan. 2013.

[22] L. Gong *et al.*, "Efficient resource allocation for all-optical multicasting over spectrum-sliced elastic optical networks," *J. Opt. Commun. Netw.*, vol. 5, pp. 836–847, Aug. 2013.

[23] V. Dukic *et al.*, "Beyond the mega-data center: networking multi-data center regions," in *Proc. of SIGCOMM 2020*, pp. 765–781, Jul. 2020.

[24] Z. Zhu *et al.*, "Build to tenants' requirements: On-demand application-driven vSD-EON slicing," *J. Opt. Commun. Netw.*, vol. 10, pp. A206–A215, Feb. 2018.

[25] S. Liu *et al.*, "DL-assisted cross-layer orchestration in software-defined IP-over-EONs: From algorithm design to system prototype," *J. Lightw. Technol.*, vol. 37, pp. 4426–4438, Sept. 2019.

[26] C. Natalino *et al.*, "Experimental study of machine-learning-based detection and identification of physical-layer attacks in optical networks," *J. Lightw. Technol.*, vol. 37, pp. 4173–4182, Aug. 2019.

[27] B. Li, W. Lu, S. Liu, and Z. Zhu, "Deep-learning-assisted network orchestration for on-demand and cost-effective VNF service chaining in inter-DC elastic optical networks," *J. Opt. Commun. Netw.*, vol. 10, pp. D29–D41, Oct. 2018.

[28] M. Salani, C. Rottondi, and M. Tornatore, "Routing and spectrum assignment integrating machine-learning-based QoT estimation in elastic optical networks," in *Proc. of INFOCOM 2019*, pp. 1738–1746, Apr. 2019.

[29] M. Furdek *et al.*, "Machine learning for optical network security monitoring: A practical perspective," *J. Lightw. Technol.*, vol. 38, pp. 2860–2871, Jun. 2020.

[30] H. Fang *et al.*, "Predictive analytics based knowledge-defined orchestration in a hybrid optical/electrical datacenter network testbed," *J. Lightw. Technol.*, vol. 37, pp. 4921–4934, Oct. 2019.

[31] B. Li, W. Lu, and Z. Zhu, "Deep-NFVOrch: leveraging deep reinforcement learning to achieve adaptive vNF service chaining in DCI-EONs," *J. Opt. Commun. Netw.*, vol. 12, pp. A18–A27, Jan. 2020.

[32] Q. Li *et al.*, "Scalable knowledge-defined orchestration for hybrid optical/electrical datacenter networks," *J. Opt. Commun. Netw.*, vol. 12, pp. A113–A122, Feb. 2020.

[33] P. Bosshart *et al.*, "P4: Programming protocol-independent packet processors," *ACM SIGCOMM Comput. Commun. Rev.*, vol. 44, pp. 87–95, Jul. 2014.

[34] H. Zhou and G. Wornell, "Efficient homomorphic encryption on integer vectors and its applications," in *Proc. of ITA 2014*, pp. 1–9, Feb. 2014.

[35] S. Tang *et al.*, "Sel-INT: A runtime-programmable selective in-band network telemetry system," *IEEE Trans. Netw. Service Manag.*, vol. 17, pp. 708–721, Jun. 2020.

[36] G. Samaraweera and J. Chang, "Security and privacy implications on database systems in Big Data era: A survey," *IEEE Trans. Knowl. Data Eng.*, vol. 33, pp. 239–258, Jan. 2021.

[37] X. Chen *et al.*, "Self-taught anomaly detection with hybrid unsupervised/supervised machine learning in optical networks," *J. Lightw. Technol.*, vol. 37, pp. 1742–1749, Apr. 2019.

[38] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. MIT Press, 2016, http://www.deeplearningbook.org.

[39] S. Moosavi-Dezfooli, A. Fawzi, and P. Frossard, "DeepFool: A simple and accurate method to fool deep neural networks," in *Proc. of CVPR 2016*, pp. 2574–2582, Jun. 2016.

[40] I. Goodfellow *et al.*, "Explaining and harnessing adversarial examples," *arXiv:1412.6572*, 2014. [Online]. Available: https://arxiv.org/abs/1412.6572.

[41] Y. Yin *et al.*, "Spectral and spatial 2D fragmentation-aware routing and spectrum assignment algorithms in elastic optical networks," *J. Opt. Commun. Netw.*, vol. 5, pp. A100–A106, Oct. 2013.

[42] Open network operating system (ONOS). [Online]. Available: https://onosproject.org/.