# Privacy-Preserving Multilayer In-Band Network Telemetry and Data Analytics: For Safety, Please do not Report Plaintext Data

Xiaoqin Pan, Shaofei Tang, Siqi Liu, Jiawei Kong, Xu Zhang, Daoyun Hu, Jin Qi, and Zuqing Zhu, *Senior Member, IEEE*

***Abstract*—With the evolution of Internet infrastructure and network services, multilayer in-band network telemetry (ML-INT) and data analytics (DA) have been considered as key enabling techniques to realize real-time and fine-grained network monitoring, especially for backbone IP-over-Optical networks. However, the existing ML-INT&DA systems have privacy and security issues, because plaintext ML-INT data is reported from the data plane and gets analyzed in the control plane. In this work, we address these issues by designing a privacy-preserving ML-INT&DA system for IP-over-Optical networks. We first leverage vector homomorphic encryption (VHE) to design a lightweight encryption scheme, which overcomes the security breaches due to eavesdropping and preserves the delicate correlations buried in multi-dimensional ML-INT data. Then, we develop an effective data compression scheme to further encode the encrypted ML-INT data and make the results suitable for hash-based signature. The signature is for data certification and enables the DA in the control plane to verify the integrity of received ML-INT data. Hence, the threats from data tampering are removed. Next, we architect a deep learning (DL) model that can directly operate on encrypted ML-INT data for anomaly detection. Finally, we implement the proposed ML-INT&DA system, and experimentally demonstrate its effectiveness in a real IP over elastic optical network (IP-over-EON) testbed, whose key elements, *i.e.*, optical line system (OLS), bandwidth-variable wavelength-selective switches (BV-WSS') and programmable data plane (PDP) switches, are all commercial products.**

***Index Terms*—In-band network telemetry (INT), IP over elastic optical networks (IP-over-EONs), Multilayer networks, Deep learning (DL), Vector homomorphic encryption (VHE), Privacy-preserving network monitoring, Data analytics, Soft failures.**

## I. Introduction

NOWADAYS, the rapid development of emerging services, such as 5G, Big Data and cloud computing, has made backbone networks increasingly complicated and highly dynamic [1]. Meanwhile, the wide deployment of virtualization technologies (*e.g.*, virtual network embedding (VNE) [2, 3] and network function virtualization (NFV) [4, 5]) has increased the difficulty of detecting and tracing down network failures, especially the soft ones [6, 7]. All these added up to cumulative stressing the monitoring and managing of the multilayer architecture of backbone networks (*i.e.*, IP-over-Optical) over time [8]. The major hassle is that the traditional techniques, such as SNMP [9] and NetFlow [10], can hardly achieve real-time and fine-grained network monitoring, which has already become one prerequisite for the network control and management (NC&M) of today's Internet.

The aforementioned dilemma can be mostly relieved by the in-band network telemetry (INT) technique [11], which has recently been promoted by the advances on programmable data plane (PDP) [12, 13]. Specifically, PDP provides network operators the flexibility to customize packet processing behaviors in the data plane, and this enables INT, which embeds telemetry data collection in packet processing pipelines for end-to-end monitoring. A typical INT system works as follows for a packet network. When a packet first enters a network with INT-based monitoring, the ingress switch inserts the preset INT instructions (*i.e.*, what to monitor and how) in it as header fields. Then, each intermediate switch checks the fields, executes the desired INT operations, and pushes the obtained telemetry data into the packet's header, as it transits the network. Finally, when the packet is about to leave the network, the egress switch pops all the telemetry data from its header, aggregates the results, and sends them to a data analyzer for real-time and fine-grained network monitoring.

Therefore, INT opens limitless possibilities for NC&M, allowing network operators to capture and identify temporary issues that emerge due to various types of failures, *i.e.*, both hard and soft ones. Following the trend, people has extended the applications of INT from packet-based single-layer networks to multilayer IP-over-Optical ones, and designed several multilayer INT and data analytics (ML-INT&DA) schemes [14–17] to facilitate real-time, fine-grained and programmable NC&M. Specifically, the ML-INT provides network operators a powerful tool to visualize both IP and optical layers in realtime, while the DA leverages deep learning (DL) to analyze rich telemetry data from both layers, for application-aware service provisioning and accurate and timely troubleshooting. Although these ML-INT&DA proposals are promising, they all overlooked the important issues related to privacy and security, because plaintext telemetry data is reported from the data plane and gets analyzed in the control plane.

The necessity of privacy-preserving ML-INT&DA systems is multiple-fold. Firstly, ML-INT provides rich telemetry data regarding a backbone network, which can be analyzed to derive sensitive information regarding the configuration and

X. Pan, S. Tang, S. Liu, J. Kong, X. Zhang, and Z. Zhu are with the School of Information Science and Technology, University of Science and Technology of China, Hefei, Anhui 230027, P. R. China (email: zqzhu@ieee.org).

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

D. Hu and J. Qi are with the Zhongxing Telecommunication Equipment (ZTE) Corporation, Nanjing, Jiangsu 210000, P. R. China.

Manuscript received on March 27, 2020.

operation of the network. In today's Internet, telemetry data usually gets reported to the control plane through control channels based on the transport layer security (TLS) or secure socket layer (SSL) connections, which are vulnerable to the man-in-the-middle attack [18]. Hence, if a malicious party taps the control channels and obtains plaintext telemetry data, it can launch various attacks based on the derived information. For instance, the operation margin of the optical layer can be obtained by analyzing the data regarding power-level and optical signal-to-noise-ratio (OSNR), and the malicious party can inject jamming or interference signals in the fiber links on which the margins for quality-of-transmission (QoT) guarantee are relatively small to amplify its attack efficiency [17].

Secondly, in additional to passive eavesdropping, the malicious party could be more aggressive to modify the telemetry data for misleading the DA system, and this would severely disturb the network automation in its network. Note that, it is known that the neural networks for DL are vulnerable to adversarial samples, which are hard to be detected and can easily cause DL to make incorrect decisions [19]. Such tampering-based attacks have already been demonstrated in [20, 21] to make IP-over-Optical networks behave strangely. Last but not least, plaintext telemetry data should not be disclosed for the consideration of privacy, if the operator wants to outsource the DL model for DA to a third party, *i.e.*, leveraging the "machine-learning-as-a-service (MLaaS)" [22] to overcome its shortage on the labor/hardware/software resources to design and train a sophisticated DL model.

As an ML-INT&DA system usually relies on the software-defined networking (SDN) architecture [23], the security and privacy issues mentioned above could be relieved if we add encryption/decryption at both ends of each data-reporting channel between the data and control planes. However, this will bring in additional operation complexity and thus processing latency, especially to the control plane. Moreover, the DA still processes plaintext telemetry data in the control plane, there are still security breaches. Note that, different from the DA in the control plane, the ML-INT in the data plane is usually handled in a distributed manner. Hence, adding data encryption in each performance monitor would not cause excessive processing burdens. Then, it would be promising if the DA can directly operate on encrypted telemetry data with its DL model. Nevertheless, most encryption schemes will just break the correlations buried in telemetry data and thus make privacy-preserving DA infeasible. More importantly, the encryption schemes cannot address the tampering-based attacks that might occur in data reporting channels.

In this work, we address the aforementioned challenges by designing a privacy-preserving ML-INT&DA system for IP-over-Optical networks. We first leverage vector homomorphic encryption (VHE) [24, 25] to design a lightweight encryption scheme, which not only preserves the delicate correlations buried in multi-dimensional ML-INT data but also limits the time complexity of data encryption. Hence, the security breaches due to eavesdropping are addressed. Then, we develop an effective data compression scheme to further encode the encrypted ML-INT data and make the results suitable for hash-based signature. With the signature, the DA in the control plane can easily verify whether encrypted ML-INT data from the data plane has been illegally modified or not. Therefore, the threats from tampering-based attacks are removed. Next, we architect a DL model that can directly operate on encrypted ML-INT data for anomaly detection.

We implement the proposed ML-INT&DA system, and experimentally demonstrate its effectiveness on privacy-preserving in a real IP over elastic optical network (IP-over-EON) testbed, whose key elements, *i.e.*, optical line system (OLS), bandwidth-variable wavelength-selective switches (BV-WSS') and PDP switches, are all commercial products. Experimental results confirm that the encryption hides sensitive information about data plane well, the encoding and signature scheme helps to detect illegally-revised data accurately, and our DL model can classify encrypted multi-dimensional ML-INT data to find the root causes of exceptions.

The rest of the paper is organized as follows. Section II briefly surveys the related work. We describe the architecture and operation principle of our privacy-preserving ML-INT&DA system in Section III, and the implementation details are in Section IV. We discuss experimental demonstrations in Section V. Finally, Section VI summarizes the paper.

## II. Related work

Recently, optical networks have been greatly impacted by the rapid evolution of SDN [26–28] and artificial intelligence (AI) [29], which promotes the idea of knowledge-defined networking (KDN) [30, 31] for network automation. Therefore, numerous studies have been focused on realizing AI-assisted network automation for optical networks [32–37]. However, one can never close the loop of automatic NC&M without realtime and fine-grained network monitoring and troubleshooting [6, 38, 39]. Traditional monitoring schemes for packet networks (*e.g.*, SNMP [9] and NetFlow [10]) utilize the out-of-band scenario based on a server-client model. Specifically, they leverage a centralized monitor to poll network elements (NEs) periodically for collecting status data. Nevertheless, the polling-based data collection cannot visualize networks in realtime or realize fine-grained monitoring to reveal the end-to-end information of arbitrary flows. Same issues also apply to traditional optical performance monitoring schemes [40–42].

The issues with traditional network monitoring schemes can be addressed with the INT technique [11], which boosted up the research and development on in-band network monitoring [43]. Consequently, people have quickly expanded the idea to consider multilayer IP-over-Optical networks and developed a few ML-INT schemes [14, 15]. Meanwhile, the INT technique has been further optimized in [44–46] to reduce its bandwidth overheads and data processing burdens. However, none of these enhanced INT versions has addressed the privacy and security issues due to eavesdropping and data tampering.

Following the idea of KDN, people have also studied how to integrate INT with DA to facilitate AI-assisted network automation. Hyun *et al.* [47] considered how to realize a self-driving network by combining INT, DL and SDN. The authors of [48] presented an architecture, namely, NetworkAI, to enable self-learning control strategies in SDN with the assistance of INT. Leveraging INT to provision latency-aware virtual network functions in metro networks has been demonstrated in [49]. Previously, in [8, 16], we proposed an ML-INT&DA system, and demonstrated real-time, fine-grained and programmable NC&M in a multilayer IP-over-Optical network testbed, where the IP layer was based on PDP switches and the optical layer was a flexible-grid EON [50–53]. Nevertheless, none of the aforementioned studies considered the privacy and security issues.

To the best of our knowledge, the privacy-preserving ML-INT&DA system, which can properly address the potential threats of eavesdropping and data tampering, has not been studied in the literature yet. This motivates us to address the problem in this work and to extend our ML-INT&DA system demonstrated in [8, 16] to a privacy-preserving one.

## III. System Architecture and Operation Principle

In this section, we describe the overall architecture of our privacy-preserving ML-INT&DA system, explain its operation principle, and elaborate on the new modules introduced in this work over the existing system developed in [8, 16].

### A. *ML-INT&DA*

Fig. 1 shows the system architecture of the proposed privacy-preserving ML-INT&DA system. The data plane is an IP-over-Optical network, where the optical layer is built with optical cross-connects (OXCs) and fiber links, and the IP layer consists of programmable data plane (PDP) switches [12, 13], client hosts, application servers, and data collection agents. The optical performance monitor (OPM) on each OXC collects telemetry data regarding the lightpaths switched by it. To monitor OSNR, power-level and spectral shape, one can leverage optical spectrum analysis [40], while more sophisticated OPM, such as the monitoring on bit-error rate (BER) and dispersion, can be accomplished by utilizing the digital signal processing modules in optical transponders [42]. As we focus on privacy-preserving in this work, we do not specify the actual scheme of OPM. In other words, telemetry data can be encrypted and processed with our proposal, no matter how it was collected.

As explained in [16], the telemetry data collected by the OPM is then sent to the local PDP switch of its OXC, by a homemade agent. The PDP switch encodes the received telemetry data regarding the optical layer together with that it collects locally about the IP layer as INT fields, and inserts them in the packets of related flows. Since the INT fields carry the telemetry data about each electrical/optical network element (NE) on a flow's routing path, real-time and fine-grained ML-INT has been realized. Note that, our ML-INT scheme also has the capability of selecting only a small portion of a flow's packets to insert the INT fields, for reducing the bandwidth overheads of ML-INT [16]. Finally, before a packet with INT fields leaves the IP-over-Optical network and reaches to its destination host, the egress PDP switch pops all the INT fields from the packet and sends them to a data collection agent, where the INT fields will be parsed, aggregated and processed to get the ML-INT data (as shown in Fig. 1).

There could be multiple data collection agents residing in the data plane, and they forward ML-INT data to the centralized controller through data reporting channels. Following the principle of KDN [30], the controller leverages DL to analyze the ML-INT data (*i.e.*, DA) and realize anomaly detection and other NC&M tasks, for AI-assisted network automation. Note that, since the IP-over-Optical network is a backbone network, it can cover a relatively large geographical area. Hence, the controller and the data collection agents normally reside at different locations, which makes the data reporting and processing vulnerable to eavesdropping and data tampering if the ML-INT data is in plaintext.

### B. *Privacy-Preserving Operations*

In order to address the privacy and security issues caused by using plaintext ML-INT data, we, in this work, design and implement a few new functional modules in the ML-INT&DA system to make its operations privacy-preserving. Specifically, the new functional modules are for both the data collection agent and the centralized controller, as shown in Fig. 1.

After obtaining ML-INT data from received INT fields, the data collection agent organizes each set of data with the same time-stamp as a plaintext vector ($\mathbf{x}$). For instance, a plaintext vector can contain a set of ML-INT data that includes the packet forwarding latency and input/output bandwidth of a PDP switch, and the input power and OSNR of the related optical port on the switch's local OXC. Since raw telemetry data may contain outlier samples, which are due to collection errors and data corruption, we introduce an outlier detection module to remove them. The outlier detection is designed based on the well-known density-based clustering algorithm (DBSCAN) [54], because the ML-INT data may distribute in irregular shapes. As we will show later in Section V, the outlier detection not only maintains the quality of ML-INT data, but also improves the robustness of our privacy-preserving DA.

Then, the plaintext vector $\mathbf{x}$ gets encrypted in the VHE-based encryption module, whose detailed operations will be described in Section IV-A. The VHE-based encryption transforms $\mathbf{x}$ into a ciphertext vector $\mathbf{c}'$. To ensure privacy-preserving, we design the VHE-based encryption to map each unique plaintext vector to different ciphertext vectors. Hence, it would be even more difficult for a malicious party to guess the plaintext vectors based on their ciphertext counterparts. More importantly, the VHE-based encryption preserves the inner correlations of ML-INT data, such that the DL-based DA in the controller can directly analyze the ciphertext vectors for anomaly detection and other NC&M tasks.

Next, to address tampering-based attack, the encoding and signature module is introduced to generate a digital signature for each ciphertext vector. It is known that hash-based signature method has acceptable time complexity and can provide sufficient data certification strength. Therefore, we develop an effective data compression scheme to further encode the ciphertext vectors and make the results suitable for hash-based signature. We will elaborate on the operation principle of the encoding and signature module in Section IV-B. Finally, the data assembly module simply appends the hash-based

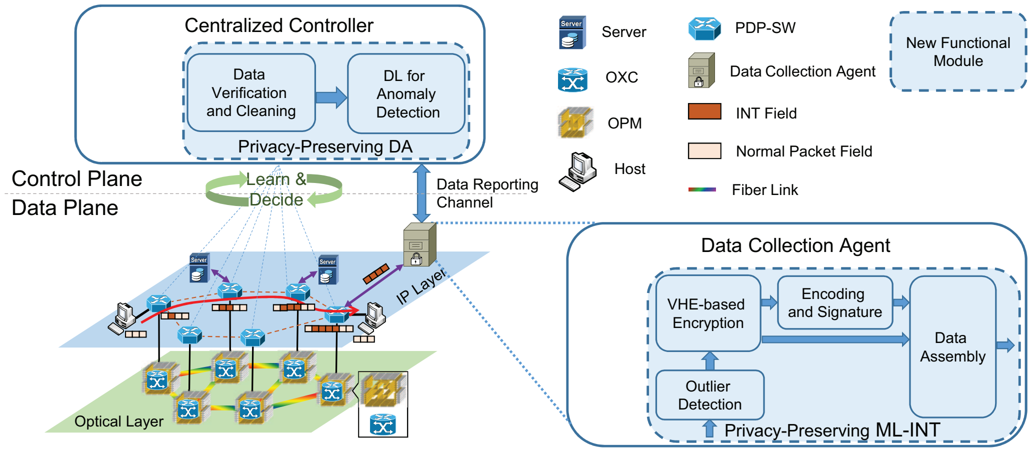


Fig. 1. Overall architecture of privacy-preserving ML-INT&DA system, PDP-SW: PDP switch, OXC: optical cross-connect, OPM: optical performance monitor.

signature to each ciphertext vector, and then forwards it to the controller through the data reporting channel.

On the other hand, the controller realizes privacy-preserving DA with the data verification and cleaning module and the DL for anomaly detection, as shown in Fig. 1. Both modules directly operate on encrypted ML-INT data (*i.e.*, the ciphertext vectors). By repeating the encoding and signature procedure discussed above and comparing the obtained signature with the one in the received message, the data verification and cleaning module verifies the integrity of ML-INT data. Specifically, if the signatures do not match, the module will classify the data as suspicious, drop it, and flag an alert if necessary. Then, trusted ML-INT data is forwarded to the DL module for anomaly detection[1], which not only determines whether the encrypted ML-INT data indicates exceptions, but also classifies the exceptions. The design and training of the DL module will be discussed in Section IV-C.

Based on the output of the DL module, the controller makes proper NC&M decisions to address the detected exceptions. In this work, we consider four types of exceptions, which are packet congestion and PDP switch misconfiguration in the IP layer, and excessive power loss and OSNR degradation in the optical layer. Meanwhile, we hope to point out that this setting is only for the purpose of experimental demonstrations, and as the privacy-preserving scheme has good universality on high-dimensional data, our proposal can handle more types of ML-INT data and/or exceptions with only minor modifications.

## IV. Implementations of Key Functional Modules

This section presents the implementation details of three key functional modules in our proposed ML-INT&DA system (*i.e.*, the VHE-based encryption, the encoding and signature, and the DL-based anomaly detection).

### A. *VHE-based Encryption*

As shown in Fig. 2, our VHE-based encryption involves three phases, *i.e.*, key generation, key-switching, and vector encryption. Initially, we have the plaintext vector for an ML-INT data sample as $\mathbf{x} \in \mathbb{Z}^m$, where all the elements in $\mathbf{x}$ are integers, and $m$ denotes the length of the vector. Meanwhile, we define $\mathbf{c} \in \mathbb{Z}^n$ as the ciphertext vector of $\mathbf{x}$ with $n \geq m$. Then, the VHE-based encryption satisfies [24, 25]

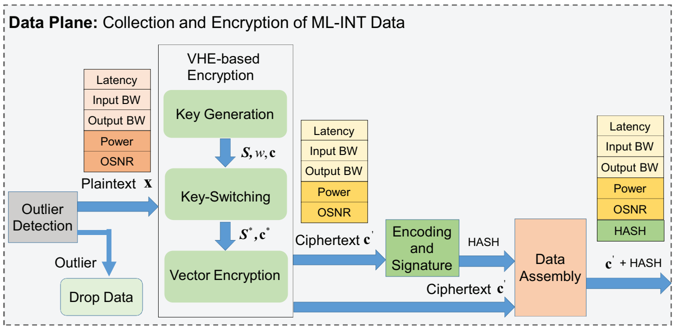


Fig. 2. Operation of privacy-preserving functional modules in data plane.

$$\mathbf{S} \cdot \mathbf{c} = w \cdot \mathbf{x} + \mathbf{e}, \tag{1}$$

where $\mathbf{S} \in \mathbb{Z}^{m \times n}$ is the private key, $\mathbf{e} \in \mathbb{Z}^m$ is the randomly-generated noise vector, and $w$ is the preset weight to balance the importance of data and noise in the encryption.

The key generation provides the private key $\mathbf{S}$ and the noise vector $\mathbf{e}$ for the vector encryption with Eq. (1). Let $\mathbf{I}$ be an $m \times m$ identity matrix. We can easily verify that $w \cdot \mathbf{x} = \mathbf{I} \cdot (w \cdot \mathbf{x})$. Hence, $\mathbf{I}$ can be the initial private key $\mathbf{S}$, which encrypts $\mathbf{x}$ as a ciphertext vector $\mathbf{c} = w \cdot \mathbf{x}$ with the noise vector as $\mathbf{e} = \mathbf{0}$.

Then, to improve the strength of the encryption, we leverage the key-switching technique. Apparently, for another pair of private key $\mathbf{S}^*$ and ciphertext vector $\mathbf{c}^*$, if they satisfy

$$\mathbf{S}^* \cdot \mathbf{c}^* = \mathbf{S} \cdot \mathbf{c}, \tag{2}$$

the relation in Eq. (1) still holds. We choose $\ell$ such that

$$\max_{i \in [1,n]} (|c_i|) < 2^{\ell}, \tag{3}$$

where $c_i$ is the $i$-th element in $\mathbf{c}$, and construct ciphertext vector $\mathbf{c}^*$ by transforming each element in $\mathbf{c}$ into the bipolar representation. Specifically, if we have $c_i = b_{i0} + b_{i1} \cdot 2 + \cdots +$

[1] Note that, with a proper design, the privacy-preserving DA can leverage the encrypted ML-INT data to accomplish other NC&M tasks too.

$b_{i(\ell-1)} \cdot 2^{\ell-1}$, $c_i$ is transformed into $\mathbf{c}_i^* = [b_{i0}, b_{i1}, \cdots, b_{i(\ell-1)}]$, where $b_{ik} \in \{-1, 0, 1\}$. Then, we obtain $\mathbf{c}^* = [\mathbf{c}_1^*, \cdots, \mathbf{c}_n^*]^T$. For instance, the vector $c = [2, -1]^T$ can be transformed into $\mathbf{c}^* = [0, 1, 0, 0, -1, 0, 0, 0]^T$, if we have $\ell = 4$. Similarly, the private key $\mathbf{S}^*$ can be obtained by converting each element $S_{ij} \in \mathbf{S}$ into a binary-related vector, *i.e.,* $\mathbf{S}_{ij}^* = [S_{ij}, 2 \cdot S_{ij}, \cdots, 2^{\ell-1} \cdot S_{ij}]$. Then, the following equality holds

$$\mathbf{S}_{ij}^* \cdot \mathbf{c}_j^* = [S_{ij}, 2 \cdot S_{ij}, \cdots, 2^{\ell-1} \cdot S_{ij}] \cdot \begin{bmatrix} b_{j0} \\ b_{j1} \\ \vdots \\ b_{j(\ell-1)} \end{bmatrix} = S_{ij} \cdot c_j. \quad (4)$$

Therefore, we construct a ciphertext vector $\mathbf{c}^*$ whose elements have their maximum absolute value as 1 and a private key $\mathbf{S}^* \in \mathbb{Z}^{m \times n \cdot \ell}$, and ensure that they satisfy Eq. (2).

Next, the vector encryption gets the final private-key-and-ciphertext pair (*i.e.,* $\mathbf{S}'$ and $\mathbf{c}'$) based on $\mathbf{S}^*$ and $\mathbf{c}^*$. We have $\mathbf{c}' \in \mathbb{Z}^{n'}$ and $\mathbf{S}' \in \mathbb{Z}^{m \times n'}$, where $n' > n$ is the predefined length of the final ciphertext vector $\mathbf{c}'$. With the procedure in [24, 25], we first construct an integer matrix $\mathbf{M} \in \mathbb{Z}^{n' \times n \cdot \ell}$ as

$$\mathbf{M} = \begin{bmatrix} \mathbf{S}^* + \mathbf{E} - \mathbf{T} \cdot \mathbf{A} \\ \mathbf{A} \end{bmatrix}, \quad (5)$$

where $\mathbf{E} \in \mathbb{Z}^{m \times n \cdot \ell}$, $\mathbf{A} \in \mathbb{Z}^{(n'-m) \times n \cdot \ell}$ and $\mathbf{T} \in \mathbb{Z}^{m \times (n'-m)}$ are all randomly-generated integer matrices. Then, we construct $\mathbf{S}'$ as $\mathbf{S}' = [\mathbf{I}, \mathbf{T}]$, where $\mathbf{I}$ is an identity matrix, and can verify

$$\mathbf{S}' \cdot \mathbf{M} = \mathbf{S}^* + \mathbf{E}. \quad (6)$$

Therefore, the final ciphertext $\mathbf{c}'$ should be

$$\mathbf{c}' = \mathbf{M} \cdot \mathbf{c}^*. \quad (7)$$

It will be easy to verify

$$\mathbf{S}' \cdot \mathbf{c}' = \mathbf{S}^* \cdot \mathbf{c}^* + \mathbf{E} \cdot \mathbf{c}^* = w \cdot \mathbf{x} + \mathbf{E} \cdot \mathbf{c}^*, \quad (8)$$

which also satisfy Eq. (1) with $\mathbf{e} = \mathbf{E} \cdot \mathbf{c}^*$.

*Algorithm* 1 illustrates the procedure of the VHE-based encryption used in our privacy-preserving ML-INT&DA system. *Line* 1 is for the initialization. The key generation is realized with *Lines* 2-3, the key-switching is accomplished by *Lines* 4-6, and the vector encryption is achieved with *Lines* 7-11. The strength of the VHE-based encryption is guaranteed by matrix $\mathbf{M}$, based on the hardness assumption of the extended learning with error problem [55]. Hence, it would be extremely difficult for an attacker to decrypt the ciphertext telemetry data within a reasonable amount of computing time. The time complexity of *Algorithm* 1 is $O(m + \ell)$, *i.e.,* a linear-time algorithm.

### *B. Data Compression Encoding for Hash-based Signature*

Fig. 2 indicates that the ciphertext vector $\mathbf{c}'$ generated by the VHE-based encryption needs to be processed by the encoding and signature module to obtain a hash-based signature for data certification. Although we can directly apply hash function to all the elements in ciphertext $\mathbf{c}'$, encoding them beforehand for data compression has the following two advantages.

First of all, as the elements in $\mathbf{c}'$ have different ranges of values, their binary representations occupy variable-length fields, which will bring additional complexity for organizing the data bytes. Therefore, if we encode the elements and make the results fit into fixed-length fields, the aforementioned hassle can be avoided. Secondly, applying hash function directly to the elements in $\mathbf{c}'$ will make the obtained signature too sensitive to the changes on $\mathbf{c}'$. In other words, the signature will not be identical, even if there is only a very minor change on one element in $\mathbf{c}'$. Note that, there could be minor changes on $\mathbf{c}'$, which were not caused by data tampering (*e.g.,* bit errors due to the noise in data transmission), and even if there is data tampering, minor modifications might not affect the DL model in the controller. Hence, if we drop all the non-identical ciphertext vectors, it is overkill and the efficiency of ML-INT&DA would be impacted. A proper data compression on $\mathbf{c}'$ can improve the signature's tolerance to minor changes.

**Algorithm 1:** VHE-based Encryption

```
   Input: weight $w$, plaintext vector $\mathbf{x}$.
   Output: ciphertext vector $\mathbf{c}'$, private key $\mathbf{S}'$.
1  get $m$ by checking $\mathbf{x}$ and set $n = m$ and $n' = n + 1$;
2  initialize private key $\mathbf{S}$ as an $m \times m$ identity matrix;
3  get initial ciphertext vector $\mathbf{c} = w \cdot \mathbf{x}$;
4  obtain the value of $\ell$ with Eq. (3);
5  transform $\mathbf{c}$ into the bipolar representation $\mathbf{c}^*$;
6  convert $\mathbf{S}$ into the binary-related vector $\mathbf{S}^*$;
7  generate a random matrix $\mathbf{T} \in \mathbb{Z}^{m \times (n'-m)}$;
8  get the final private key $\mathbf{S}' = [\mathbf{I}, \mathbf{T}]$;
9  get random matrices $\mathbf{E} \in \mathbb{Z}^{m \times n \cdot \ell}$ and $\mathbf{A} \in \mathbb{Z}^{(n'-m) \times n \cdot \ell}$;
10 construct $\mathbf{M}$ with Eq. (5);
11 calculate the final ciphertext vector $\mathbf{c}'$ with Eq. (7);
12 return($\mathbf{c}'$, $\mathbf{S}'$);
```

Inspired by the nonuniform quantization in speech communications [56], we design the data compression scheme as explained in *Algorithm* 2. The data compression will encode each element in a ciphertext vector $\mathbf{c}'$ with one byte[2]. As illustrated in Fig. 3, the byte is divided into two portions, *i.e.,* the paragraph code with $n$ bits and the level code with $(8-n)$ bits. Here, the paragraph code represents which nonuniform region the element $c_i$ stays in, while the level code denotes the result of the uniform quantization for $c_i$ in the region. In *Algorithm* 2, *Line* 1 is for the initialization to normalize $c_i$ as a non-negative value $\hat{c}_i \in [0, c_i^{\max} - c_i^{\min}]$. Then, the nonuniform regions for quantization are obtained with *Lines* 2-7, while the data compression is accomplished with *Lines* 8-12.

Note that, the value of $n$ (*i.e.,* the length of the paragraph code field) determines the tolerance of the change on the original element $c_i$, which is simply the maximum quantization error caused by the data compression. For instance, if we set $n = 3$ bits, the tolerance changes within $[\frac{1}{8192}, \frac{1}{128}] \cdot (c_i^{\max} - c_i^{\min})$, depending on which region the element is actually in. Therefore, the value of $n$ can be determined empirically based on the dynamic range of $c_i$, and in the extreme case with $n = 0$, the data compression uses uniform quantization. As we always have $n < 8$, *Algorithm* 2 is constant time (*i.e.,* its

[2] Note that, the data compression is just for the hash-based signature, while the ciphertext vector $\mathbf{c}'$ will still be sent to the controller without compression.

**Algorithm 2:** Data Compression for Hash-based Signature

```
Input: an element $c_i$ in ciphertext $\mathbf{c}'$, upper- and
       lower-bounds of $c_i$ ($c_i^{\max}$ and $c_i^{\min}$), length of
       paragraph code field in bits ($n$).
Output: one-byte encoding of $c_i$ ($b_i$).
$\hat{c}_i = c_i - c_i^{\min}$, $x = c_i^{\max} - c_i^{\min}$;
for $j = 0$ to $2^n - 1$ do
    if $j = 2^n - 1$ then
        $y = x$, $x = c_i^{\min}$;
    else
        $y = x$, $x = \frac{x}{2}$;
    end
    if $\hat{c}_i \in [x, y]$ then
        encode the paragraph code in $b_i$ as the binary
        representation of $j$;
        quantize $\hat{c}_i$ in $[x, y]$ uniformly with $(8 - n)$ bits
        as the level code in $b_i$;
        break;
    end
end
return($b_i$);
```

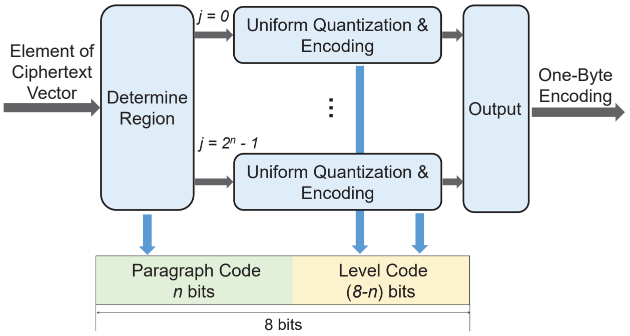


Fig. 3. Data compression based encoding for hash signature.

complexity is $O(1)$).

### C. DL Model for Anomaly Detection

Fig. 4 explains the operation of privacy-preserving functional modules in the controller. The data verification and cleaning module repeats the procedure discussed in the previous subsection, obtains a hash-based signature based on the ciphertext $\mathbf{c}'$ that it received, and checks the signature with the received one for data verification. If the signatures do not match, the ciphertext $\mathbf{c}'$ will be dropped. Next, trusted ciphertext vectors are forwarded to the DL model for anomaly detection. We design the DL model based on a deep neural network (DNN), and apply supervised learning to train it for detecting and classifying exceptions based on the encrypted ML-INT data. Note that, for a traffic flow that passes through $N$ PDP switches in the IP-over-Optical network, a complete set of the ML-INT data carried by its packets contains $N$ ciphertext vectors, for the end-to-end multilayer routing. The DL model performs anomaly detection on each complete set of ML-INT data, *i.e.*, analyzing $N$ ciphertext vectors in sequence. Hence, the DL model can not only detect the exceptions but also locate them on the packets' multilayer routing path.

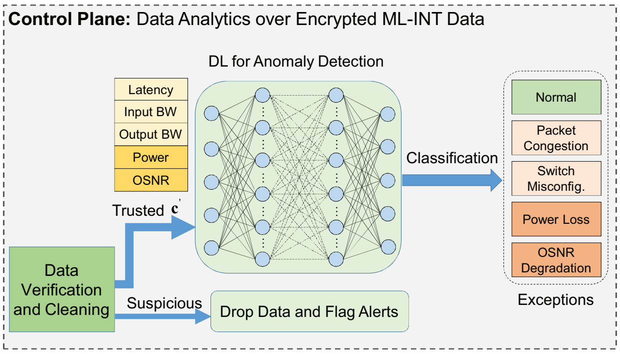


Fig. 4. Operation of privacy-preserving functional modules in control plane.

The DNN includes seven layers, which are the input layer whose neurons match with the dimension of a ciphertext vector $\mathbf{c}'$, five hidden layers with $\{256, 128, 64, 32, 16\}$ neurons, respectively, and the output layer where the number of neurons equals that of exceptions plus one (*i.e.*, for the normal case). Except for the output layer, which uses the softmax activation function, all the layers in the DNN utilize relu as their activation functions. We leverage the categorical cross-entropy function to design the loss function, and use it to describe the DNN's accuracy on the multi-class classification for anomaly detection. Then, the DNN is trained in the offline manner to minimize the loss, *i.e.*, its parameters are optimized iteratively over labeled training samples with the classic back-propagation and gradient descent algorithm [29].

## V. Experimental Demonstrations

In this section, we implement the proposed ML-INT&DA system in an IP-over-EON testbed, and conduct experiments on anomaly detection to verify and evaluate its performance. As hard failures usually cause immediate service interruptions, the techniques to detect and locate them have already been mature [40]. Therefore, the anomaly detection discussed in this section will focus on the soft failures that only induce minor degradations on network operation, *e.g.*, small reductions on OSNR and power-level, and minor misconfigurations and bandwidth shrinking on PDP switches. These soft failures do not cause relevant service disruptions immediately and thus are difficult to be detected and located, but if the operator keeps ignoring them, their impacts can accumulate over time and eventually introduce unsolvable complications [6].

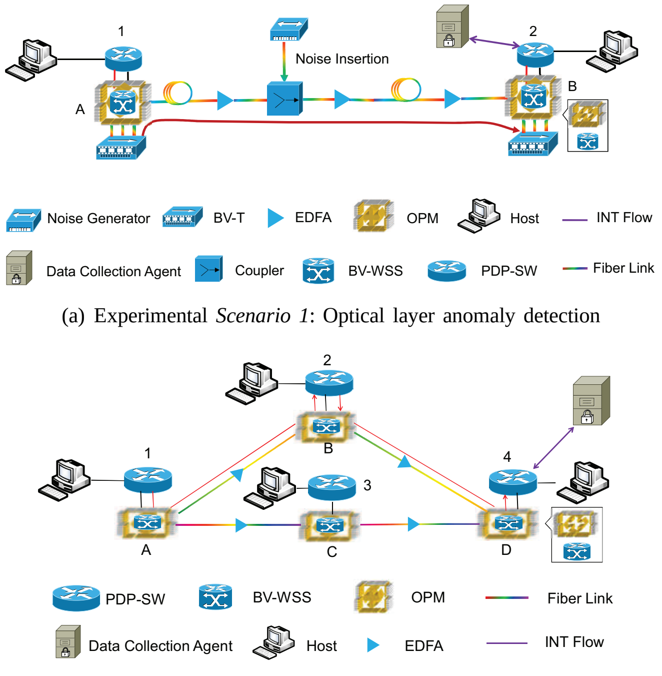
### A. Testbed Setup

*1) Data Plane:* Our testbed uses a small but real IP-over-EON as the data plane. Supporting network disaggregation [15], the optical layer is an EON that enables flexible-grid spectrum allocation and bandwidth-variable data transmission [57]. Specifically, the EON consists of two types of major elements, *i.e.*, disaggregated optical line system (OLS) and bandwidth-variable wavelength-selective switches (BV-WSS').

The disaggregated OLS is realized based on the Juniper BTI7800 platform, in which each bandwidth-variable transponder (BV-T) can leverage different modulation formats (*e.g.*, PM-QPSK and 16-QAM) to achieve a line-rate of $[100, 400]$ Gbps, and the BV-Ts are connected with fiber links with in-line erbium-doped fiber amplifiers (EDFAs). Each fiber link contains 30 km standard single-mode fiber (SSMF), and due to the shortage of SSMF in our lab, we have difficulty in using longer fiber links. Therefore, the experiments use an amplified spontaneous emission (ASE) noise generator, dispersion compensation modules, and power attenuators to emulate the effects of longer fiber transmissions. The disaggregated nature of the OLS system enables our ML-INT&DA to collect rich telemetry data regarding the optical layer, such as optical power-level, OSNR, bit-error-rate before forward error correction (BERbFEC), chromatic dispersion (CD), differential group delay (DGD), *etc*. The BV-WSS' are also commercial products, each of which has $1\times9$ configuration, operates within $[1528.43, 1566.88]$ nm, and provides a spectrum allocation granularity of $12.5$ GHz.

The IP layer of the testbed consists of PDP switches, client hosts, and data collection agents. The PDP switches are 3.2-Tbps Barefoot switches equipped with Tofino ASICs, which support P4-based network programming [12]. They have 10/40 GbE optical ports, and can be programmed to collect ML-INT data and insert them into packets as INT fields. Each host is emulated with a commercial traffic generator/analyzer that can send/receive data up to $40$ Gbps. The data collection agents are homemade, and they run on high-performance Linux servers to accomplish the collection and encryption of ML-INT data.

*2) Control Plane:* Developed based on the open network operating system (ONOS) platform [58], the control plane of our testbed leverages a centralized controller to monitor and manage the IP-over-EON. In addition to the conventional SDN-based NC&M tasks, the ONOS-based controller also communicates with the data collection agents through TCP connections to receive encrypted ML-INT data. Hence, it can perform privacy-preserving DA over the encrypted ML-INT data for anomaly detection. The controller also runs on a high-performance Linux server.



(a) Experimental *Scenario 1*: Optical layer anomaly detection

(b) Experimental *Scenario 2*: Multilayer anomaly detection

Fig. 5. Two scenarios for function verification and performance evaluation.

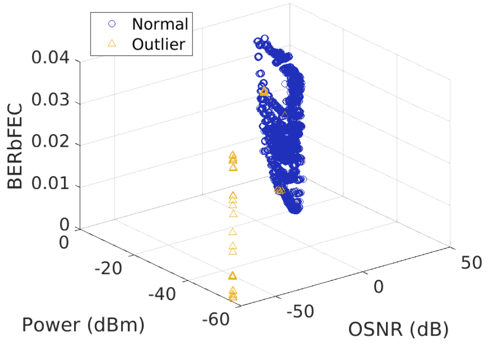


Fig. 6. Results of the outlier detection in *Scenario 1*.

## *B. Experimental Scenarios*

With the testbed, we consider two experimental scenarios to verify the performance of our proposal and demonstrate its universality. Specifically, we first use the privacy-preserving ML-INT&DA system to monitor a single hop in the EON, and then apply it to oversee both the IP and optical layers.

*1) Scenario 1: Optical Layer Anomaly Detection:* Fig. 5(a) shows the first experimental scenario, which is for optical layer anomaly detection. Here, we have a $100$ Gbps lightpath from *Node A* to *Node B*. The OLS system assigns its central wavelength as $1550.39$ nm and allocates $50$ GHz bandwidth to it (*i.e.*, four $12.5$-GHz frequency slots (FS')). As indicated in Fig. 5(a), we use an ASE noise generator to insert noise in the fiber link between the two nodes.

In the experiments, we apply different configurations to the OLS system, such that the power-level and OSNR of the lightpath exhibit various combinations at the receiver end. Then, we program our ML-INT system to collect the power-level, OSNR and BERbFEC of the lightpath. To test the performance of the privacy-preserving DA, the data collection agent organizes the telemetry data on power-level and OSNR as plaintext ML-INT vectors, and labels the vectors based on their corresponding BERbFEC. We set the threshold as $0.015$ for BERbFEC. Specifically, if the combination of power-level and OSNR in a vector corresponds to a BERbFEC that is lower than $0.015$, the vector is labeled as "Normal". Otherwise, we label the vector as "Low Power" or "Degraded OSNR" according to the major reason of the high BER value, *i.e.*, the power-level is too low or the OSNR gets degraded too much. Next, the data collection agent encrypts the ML-INT vectors and forwards them to the controller for anomaly detection.

The experiments with *Scenario 1* collect $\sim$$31,000$ ML-INT vectors as data samples. We use $90\%$ of them to train the DL model in the controller, and the remaining $10\%$ samples are included in the testing set for performance evaluation.

*2) Scenario 2: Multilayer Anomaly Detection:* The second experimental scenario is for multilayer anomaly detection, as shown in Fig. 5(b). This time, we use the proposed privacy-preserving ML-INT&DA system to supervise both the IP and optical layers of an IP-over-EON that consists of four nodes. Here, the host connecting to *PDP Switch 1* transmits a $10$ Gbps packet flow to the one on *PDP Switch 4*. The flow is routed in the IP-over-EON network as indicated by the red solid line in Fig. 5(b), *i.e.*, *PDP Switch 1→BV-WSS A→BV-WSS B→PDP Switch 2→BV-WSS B→BV-WSS D→PDP Switch 4*. Hence, the multilayer routing path involves two lightpaths in the optical layer and three PDP switches in the IP layer.

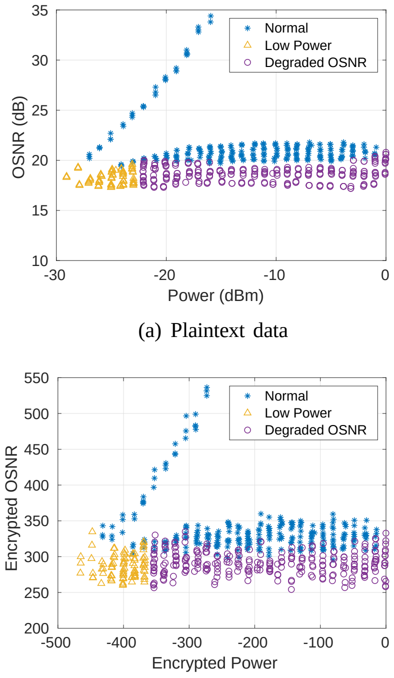
We still program our ML-INT system to collect the power-level and OSNR of each lightpath at its receiver end, and meanwhile, IP layer telemetry is realized with the PDP switches, which collect the packet processing latency and the input and output bandwidths regarding each switch port pair. Therefore, for *Scenario 2*, each ML-INT vector includes five elements. Then, with the IP-over-EON testbed in Fig. 5(b), we apply various settings (*e.g.*, various routing and spectrum assignments for lightpaths, different EDFA settings and noise insertion, various traffic routing, bandwidth usages, and flow-table configurations in the PDP switches), emulate different exception cases in the IP and optical layers, and collect and encrypt $\sim 18,000$ ML-INT vectors as data samples. For each case, we monitor the receiving bandwidth of the flow at its destination host, and flag an exception if the bandwidth is below $9$ Gbps temporarily. Then, each ML-INT vector is labeled with the actual root-cause of its exception (if there is any), which can be “Normal”, “Low Power”, “Degraded OSNR”, “Packet Congestion”, and “Switch Misconfiguration”. Similar to that in *Scenario 1*, we still put $90\%$ of the data samples in the training set of the DL model in the controller, while the remaining $10\%$ samples are included in the testing set for its performance evaluation.

### *C. Performance of Outlier Detection*

As shown in Fig. 2, the data collection agent first leverages the outlier detection module to remove the outlier samples that are due to collection errors and data corruption. Taking *Scenario 1* as an example, we plot a portion of the raw telemetry data in Fig. 6. The 3-dimensional (3D) plot indicates that the normal data samples confine to a 3D cluster, while the outliers lay far away from the cluster. Here, the “normal” samples do not mean that they correspond to normal network operations, but just suggest that they were collected without collection errors or data corruption. Apparently, to ensure the accuracy of subsequent anomaly detection, all the outliers should be detected and removed. Because normal samples confining to irregular-shaped high-dimensional clusters, we design the outlier detection based on DBSCAN, which detects $98.9\%$ outliers on average in the experiments.

### *D. Performance of Privacy-Preserving Feature*

We verify the privacy-preserving feature of our proposal as follows. Because the VHE-based encryption has similar effects on the plaintext of multi-dimensional ML-INT data and *Scenarios 1* and *2* both consider the power-level and OSNR of lightpaths, we take the correlation between them as an example, and plot the plaintext and ciphertext of them for *Scenarios 1* and *2* in Figs. 7 and 8, respectively. By comparing the plaintext and ciphertext in the figures, we can clearly see why our VHE-based encryption can preserve the privacy of data plane. The results indicate that each plaintext sample gets spread to a cluster of encrypted ones, *i.e.*, the VHE-based encryption will not encode a plaintext sample to the same value in different rounds, and the ciphertext samples use meaningless values for power-level and OSNR. This makes illegal decryption difficult, and prevents malicious parties from analyzing the data samples to derive sensitive information about the configuration and operation of the network. Meanwhile, the figures also suggest that the correlation between power-level and OSNR gets preserved through the encryption. This can be further justified, if we check the classification accuracy of the anomaly detection. Note that, an attacker can also use a DL model to get the correlations buried in the ciphertext telemetry data, but this will not bring anything good to its attacks. This is because the ciphertext data does not have any physical meaning, and thus the attacker cannot derive any sensitive information about the IP-over-Optical network.

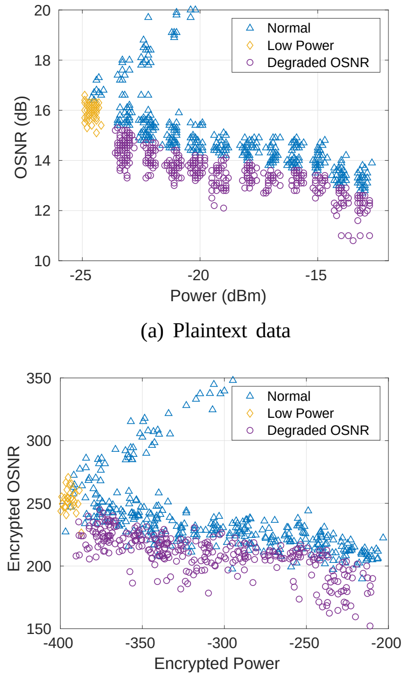


(a) Plaintext data

(b) Ciphertext data

Fig. 7. Results of VHE-based encryption in *Scenario 1*.

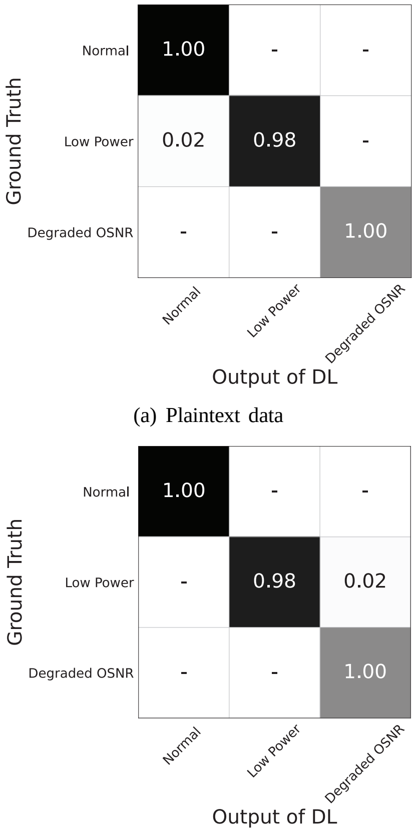
In the experiments, we train two DL models with the same structure for each scenario, to operate on plaintext and ciphertext ML-INT data, respectively. The performance of the DL-based anomaly detection is shown in Table I. The results indicate that with our designs of the VHE-based encryption and DL model, the DA over encrypted ML-INT data provides similar training/testing accuracies and uses similar training time, as that over plaintext ML-INT data. Note that, the training for the DL model in *Scenario 1* takes longer time because its training set contains more samples. This confirms the feasibility of our proposal. Finally, Figs. 9 and 10 plot the



(a) Plaintext data

(b) Ciphertext data

Fig. 8. Results of VHE-based encryption in *Scenario 2*.



(a) Plaintext data

(b) Ciphertext data

Fig. 9. Confusion matrices of DL-based anomaly detection in *Scenario 1*.

confusion matrices that describe the DL models' performance on their testing sets, for *Scenarios 1* and *2*, respectively. The figures show the breakdown of classification errors.

TABLE I
PERFORMANCE OF DL-BASED ANOMALY DETECTION

| Experimental Scenario | Accuracy on Training Set | Accuracy on Testing Set | Training Time (s) |
|---|---|---|---|
| Plaintext in *Scenario 1* | 99.26% | 99.31% | 44.33 |
| Ciphertext in *Scenario 1* | 99.07% | 99.48% | 45.30 |
| Plaintext in *Scenario 2* | 98.67% | 98.83% | 14.95 |
| Ciphertext in *Scenario 2* | 98.90% | 97.91% | 14.91 |

### *E. Performance of Protection against Tampering Attacks*

The VHE-based encryption and the DL model that can directly operate on encrypted ML-INT data address passive eavesdropping. We evaluate the encoding and signature module in the following to verify our proposal's protection against tampering-based attacks. Specifically, for the ciphertext vectors $\{\mathbf{c}'\}$, we randomly modify a fixed portion of their elements in the data reporting channel, then let the controller process them with the data verification and cleaning module and the DL model, and finally check the data tampering's effect on the classification accuracy of anomaly detection.

**Definition 1.** *We define the* ***tampering ratio*** *as the portion of the ciphertext elements that are illegally modified in the data reporting channel, which changes within* $[10\%, 50\%]$.

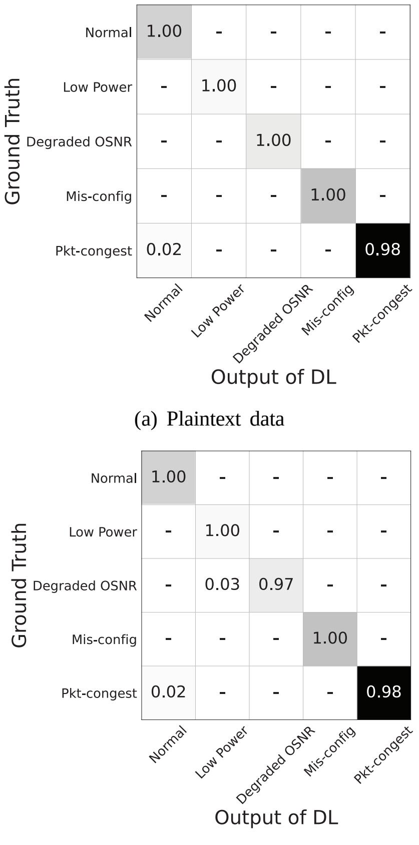
**Definition 2.** *We define* $\delta$ *as the* ***minimum tolerance*** *provided by our encoding and signature scheme.*

For instance, if the encoding scheme assigns $n = 3$ bits to the paragraph code field in Fig. 3, we have $\delta = \frac{1}{8192} \cdot (c_i^{\max} - c_i^{\min})$, where $(c_i^{\max} - c_i^{\min})$ is the dynamic range of the element $c_i$ in ciphertext vector $\mathbf{c}'$. Then, for the encoding scheme with $n = 3$, any tampering on $c_i$ that is greater than $64 \cdot \delta$ will be detected by the hash-based signature.

In the experiments, we set $n = 3$ and consider three scenarios as Minor, Moderate, and Severe tampering, with the modifications falling in $[1, 64]$, $(32, 2048]$, and $(64, 4096]$ times of $\delta$, respectively, to analyze how data tampering impacts our privacy-preserving DA. The experiments also compare the classification accuracy of the DL model, with and without the signature based data verification. The results in Figs. 11 and 12 indicate that for all the experimental scenarios, our encoding and signature scheme successfully protect the system against tampering-based attacks. Specifically, the DL's accuracies on data without tampering and tampered data with data verification are almost the same, and the tampering-based attacks only reduce the accuracies slightly. On the other hand, except for the Minor tampering cases, the data tampering can severely affect the DL's accuracy when the data verification is absent. Therefore, the results confirm that our data certification and verification with the encoding and signature scheme can detect illegally-revised data accurately and protect the ML-INT&DA system against tampering-based attacks.

### *F. Stress Tests on Data Collection Agent and Controller*

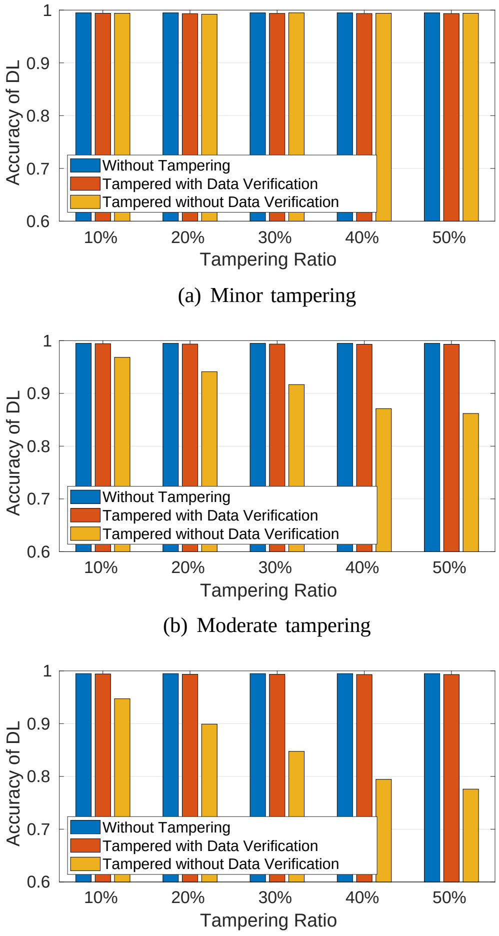
As we have explained in Sections IV-A and IV-B, the time complexities of our VHE-based encryption (*Algorithm* 1) and



(a) Plaintext data

(b) Ciphertext data

Fig. 10. Confusion matrices of DL-based anomaly detection in *Scenario 2*.



(a) Minor tampering

(b) Moderate tampering

(c) Severe tampering

Fig. 11. Results of protection against tampering attacks for *Scenario 1*.

data compression based encoding (*Algorithm* 2) are linear and constant, respectively. Therefore, they will run very fast in practical NC&M systems and will not cause any scalability issues. To verify this, we conduct two stress tests to evaluate the computing overheads that the algorithms will bring in. As the plaintext vectors in *Scenario 2* includes more elements, we perform the stress tests with them.

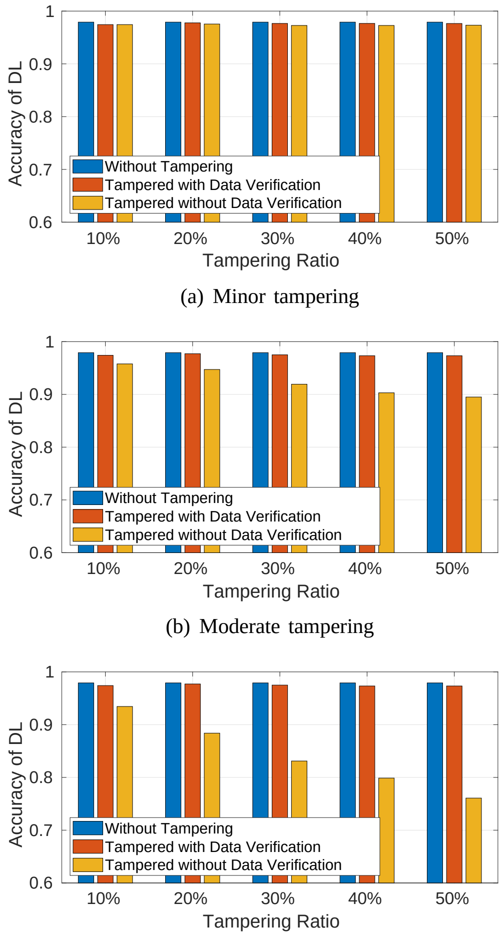
In the first test, we stress the data collection agent by letting it encrypt, encode and sign a large amount of ML-INT vectors within a second to prepare a batch report for sending to the controller. Note that, we design the data collection agent such that these privacy-preserving tasks get handled in parallel with the agent's normal operations of parsing, aggregating and processing INT fields in packets to get and record ML-INT data. The total processing time to prepare the batch report is shown in Fig. 13(a), which indicates that the data collection agent can finish the whole data processing within $2.4$ seconds, when it gets $10,000$ ML-INT vectors within a second to report to the controller. The results confirm that our algorithms are lightweight, and the encoding and signature runs much faster than the VHE-based encryption.

Note that, an ML-INT&DA system normally should not send telemetry data to the controller in the constant manner, because this will flood the controller and prevent it from conducting other NC&M tasks. Hence, except for reporting urgent issues, each data collection agent uses a polling interval of at least tens of seconds. To this end, we can see that the data collection agent is suitable for practical network operations.

In the second test, we flood thousands of ML-INT vectors to the controller within a second, wait it to process all the data, and measure the total processing time and the CPU usage on the server that runs the controller. Here, the total processing time includes both the time for the communication to report the data and the time used to process them. Fig. 13(b) shows that the total processing time is almost identical when the data collection agent reports plaintext and encrypted ML-INT vectors to the controller. The CPU usages in Fig. 13(c) suggest that the DA over encrypted data only costs slightly more CPU usage ($\sim$$0.2\%$). Hence, the results further confirm that our privacy-preserving scheme would not cause noticeable overhead or slow down the processing in the control plane.

## VI. Conclusion

In this paper, we designed and experimentally demonstrated a privacy-preserving ML-INT&DA system for IP-over-Optical networks. We first developed a lightweight VHE-based encryption scheme to encrypt ML-INT data, such that the security breaches due to eavesdropping can be addressed and the inner correlations buried in the multi-dimensional ML-INT data can be preserved. Then, we designed an effective data compression scheme to further encode the encrypted ML-INT

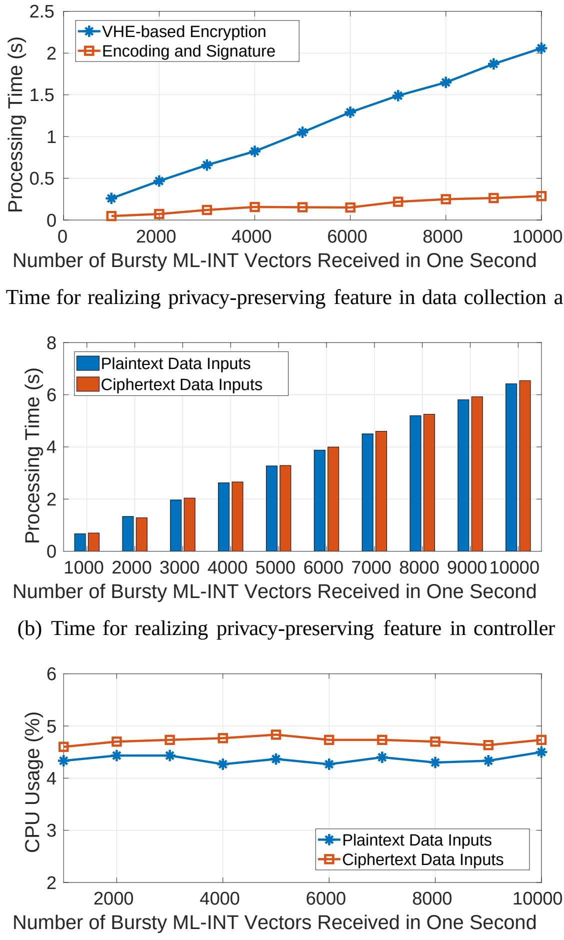


(a) Minor tampering

(b) Moderate tampering

(c) Severe tampering

Fig. 12. Results of protection against tampering attacks for *Scenario 2*.



(a) Time for realizing privacy-preserving feature in data collection agent

(b) Time for realizing privacy-preserving feature in controller

(c) CPU usage for realizing privacy-preserving feature in controller

Fig. 13. Results of stress tests with ML-INT data from *Scenario* 2.

data and make the results suitable for hash-based signature, which enables the DA in the control plane to easily verify the integrity of received ML-INT data. Next, we architected a DL model that can directly operate on the encrypted ML-INT data for anomaly detection. Finally, we implemented the proposed ML-INT&DA system, and demonstrated it experimentally in a real IP-over-EON testbed built with commercial network elements. The experimental results confirmed that our VHE-based encryption hides sensitive information regarding data plane well, the encoding and signature scheme can detect illegally-revised data and protect against tampering attacks, and our DL model can classify encrypted ML-INT data with high accuracy to find the root causes of exceptions.

## Acknowledgments

This work was supported in part by the NSFC projects 61871357, 61771445 and 61701472, ZTE Research Fund PA-HQ-20190925001J-1, Zhejiang Lab Research Fund 2019LE0AB01, CAS Key Project (QYZDY-SSW-JSC003), and SPR Program of CAS (XDC02070300).

## 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, H. Jiang, Y. Wang, and Z. Zhu, “Novel location-constrained virtual network embedding (LC-VNE) algorithms towards integrated node and link mapping,” *IEEE/ACM Trans. Netw.*, vol. 24, pp. 3648–3661, Dec. 2016.

[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] W. Fang *et al.*, “Joint spectrum and IT resource allocation for efficient vNF service chaining in inter-datacenter elastic optical networks,” *IEEE Commun. Lett.*, vol. 20, pp. 1539–1542, Aug. 2016.

[6] R. Govindan *et al.*, “Evolve or die: High-availability design principles drawn from Google’s network infrastructure,” in *Proc. of ACM SIGCOMM 2016*, pp. 58–72, Aug. 2016.

[7] J. Yin *et al.*, “Experimental demonstration of building and operating QoS-aware survivable vSD-EONs with transparent resiliency,” *Opt. Express*, vol. 25, pp. 15 468–15 480, 2017.

[8] 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.

[9] J. Case, M. Fedor, M. Schoffstall, and J. Davin, “A simple network management protocol (SNMP),” *RFC 1098*, May 1990. [Online]. Available: https://tools.ietf.org/html/rfc1157

[10] B. Claise, “Cisco systems NetFlow services export version 9,” *RFC 3954*, Oct. 2004. [Online]. Available: https://tools.ietf.org/html/rfc3954

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

[12] P. Bosshart *et al.*, “P4: Programming protocol-independent packet pro-

cessors," *ACM SIGCOMM Comput. Commun. Rev.*, vol. 44, pp. 87–95, Jul. 2014.

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

[14] M. Anand, R. Subrahmaniam, and R. Valiveti, "POINT: An intent-driven framework for integrated packet-optical in-band network telemetry," in *Proc. of ICC 2018*, pp. 1–6, Jun. 2018.

[15] F. Paolucci, A. Sgambelluri, F. Cugini, and P. Castoldi, "Network telemetry streaming services in SDN-based disaggregated optical networks," *J. Lightw. Technol.*, vol. 36, pp. 3142–3149, Aug. 2018.

[16] 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, 2019.

[17] 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.

[18] N. Dayal, P. Maity, S. Srivastava, and R. Khondoker, "Research trends in security and DDoS in SDN," *Secur. Commun. Netw.*, vol. 9, pp. 6386–6411, Dec. 2016.

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

[20] 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.

[21] 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.

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

[23] N. Xue *et al.*, "Demonstration of OpenFlow-controlled network orchestration for adaptive SVC video manycast," *IEEE Trans. Multimedia*, vol. 17, pp. 1617–1629, Sept. 2015.

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

[25] A. Yu, W. Lai, and J. Payor. (2015, May) Efficient integer vector homomorphic encryption. [Online]. Available: https://courses.csail.mit.edu/6.857/2015/files/yu-lai-payor.pdf

[26] L. Liu *et al.*, "OpenSlice: an OpenFlow-based control plane for spectrum sliced elastic optical path networks," *Opt. Express*, vol. 21, pp. 4194–4204, Feb. 2013.

[27] C. Chen *et al.*, "Demonstrations of efficient online spectrum defragmentation in software-defined elastic optical networks," *J. Lightw. Technol.*, vol. 32, pp. 4701–4711, Dec. 2014.

[28] Z. Zhu *et al.*, "Demonstration of cooperative resource allocation in an OpenFlow-controlled multidomain and multinational SD-EON testbed," *J. Lightw. Technol.*, vol. 33, pp. 1508–1514, Apr. 2015.

[29] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. MIT Press, 2016.

[30] A. Mestres *et al.*, "Knowledge-defined networking," *ACM SIGCOMM Comput. Commun. Rev.*, vol. 47, pp. 2–10, Jul. 2017.

[31] 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.

[32] L. Velasco *et al.*, "An architecture to support autonomic slice networking," *J. Lightw. Technol.*, vol. 36, pp. 135–141, Jan. 2018.

[33] 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.

[34] D. Rafique *et al.*, "Cognitive assurance architecture for optical network fault management," *J. Lightw. Technol.*, vol. 36, pp. 1443–1450, Apr. 2018.

[35] 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.

[36] 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.

[37] 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.

[38] 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.

[39] G. Liu *et al.*, "Hierarchical learning for cognitive end-to-end service provisioning in multi-domain autonomous optical networks," *J. Lightw. Technol.*, vol. 37, pp. 218–225, Jan. 2019.

[40] D. Kilper *et al.*, "Optical performance monitoring," *J. Lightw. Technol.*, vol. 22, pp. 294–304, Jan. 2004.

[41] Z. Zhu *et al.*, "Jitter and amplitude noise accumulations in cascaded all-optical regenerators," *J. Lightw. Technol.*, vol. 26, pp. 1640–1652, Jun. 2008.

[42] Z. Dong *et al.*, "Optical performance monitoring: A review of current and future technologies," *J. Lightw. Technol.*, vol. 34, pp. 525–543, Jan. 2016.

[43] C. Kim *et al.*, "In-band network telemetry via programmable dataplanes," in *Proc. of ACM SIGCOMM 2015*, pp. 1–2, Aug. 2015.

[44] Y. Kim, D. Suh, and S. Pack, "Selective in-band network telemetry for overhead reduction," in *Proc. of CloudNet 2018*, pp. 1–3, Oct. 2018.

[45] S. Tang *et al.*, "Sel-INT: A runtime-programmable selective in-band network telemetry system," *IEEE Trans. Netw. Serv. Manag., in Press*, 2019.

[46] J. Vestin *et al.*, "Programmable event detection for in-band network telemetry," *arXiv preprint arXiv:1909.12101*, 2019. [Online]. Available: https://arxiv.org/abs/1909.12101

[47] J. Hyun, V. Nguyen, and J. Hong, "Towards knowledge-defined networking using in-band network telemetry," in *Proc. of NOMS 2018*, pp. 1–7, Apr. 2018.

[48] H. Yao *et al.*, "NetworkAI: An intelligent network architecture for self-learning control strategies in software defined networks," *IEEE Internet Things J.*, vol. 5, pp. 4319–4327, Dec. 2018.

[49] F. Cugini *et al.*, "P4 in-band telemetry (INT) for latency-aware VNF in metro networks," in *Proc. of OFC 2019*, pp. 1–3, Mar. 2019.

[50] 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.

[51] 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.

[52] L. Zhang and Z. Zhu, "Spectrum-efficient anycast in elastic optical inter-datacenter networks," *Opt. Switch. Netw.*, vol. 14, pp. 250–259, Aug. 2014.

[53] X. Chen, F. Ji, and Z. Zhu, "Service availability oriented p-cycle protection design in elastic optical networks," *J. Opt. Commun. Netw.*, vol. 6, pp. 901–910, Oct. 2014.

[54] M. Ester, H. Kriegel, J. Sander, and X. Xu, "A density-based algorithm for discovering clusters in large spatial databases with noise," in *Proc. of KDD 1996*, pp. 226–231, Aug. 1996.

[55] Z. Brakerski, C. Gentry, and S. Halevi, "Packed ciphertexts in LWE-based homomorphic encryption," in *Proc. of PKC 2013*, pp. 1–13, Feb. 2013.

[56] T. Rappaport, *Wireless Communications Principles and Practice*. Prentice Hall, 1996.

[57] 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.

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