Self-Taught Anomaly Detection With Hybrid Unsupervised/Supervised Machine Learning in Optical Networks

Self-taught Anomaly Detection with Hybrid Unsupervised/Supervised Machine Learning in Optical Networks

Xiaoliang Chen, *Member, IEEE*, Baojia Li, Roberto Proietti, Zuqing Zhu, *Senior Member, IEEE*, S. J. Ben Yoo, *Fellow, IEEE, Fellow, OSA*

(Invited Paper)

***Abstract*—This paper proposes a self-taught anomaly detection framework for optical networks. The proposed framework makes use of a hybrid unsupervised and supervised machine learning scheme. First, it employs an unsupervised data clustering module (DCM) to analyze the patterns of monitoring data. The DCM enables a self-learning capability that eliminates the requirement of prior knowledge of abnormal network behaviors and therefore can potentially detect unforeseen anomalies. Second, we introduce a self-taught mechanism that transfers the patterns learned by the DCM to a supervised data regression and classification module (DRCM). The DRCM, whose complexity is mainly related to the scale of the applied supervised learning model, can potentially facilitate more scalable and time-efficient online anomaly detection by avoiding excessively traversing the original dataset. We designed the DCM and DRCM based on the density-based clustering algorithm, and the deep neural network structure, respectively. Evaluations with experimental data from two use cases (i.e., single-point detection and end-to-end detection) demonstrate that up to 99% anomaly detection accuracy can be achieved with a false positive rate below 1%.**

***Index Terms*—Self-taught anomaly detection, Hybrid unsupervised and supervised machine learning, Data clustering module (DCM), Data regression and classification module (DRCM).**

I. Introduction

EFFECTIVE fault management is vital for assuring the correct operations and required quality-of-service of (QoS) optical networks [1], [2]. Typically, network faults can be categorized into hard failures (e.g., fiber cuts), and anomalies (or soft failures), which can be caused by various factors, such as component aging and malfunctioning [3], control and management plane errors, physical layer attacks [4], etc. While hard failures can cause immediate service disruptions and can be easily detected, isolated, and restored [5]–[7], such anomalies may gradually degrade the performance of optical networks and are covert before they induce significant deviations of network parameters. Accurate and efficient anomaly detection and identification in optical networks are therefore highly desired but also challenging.

X. Chen, R. Proietti and S. J. B. Yoo are with the Department of Electrical and Computer Engineering, University of California, Davis, Davis, CA 95616, USA (Email: xlichen@ucdavis.edu, sbyoo@ucdavis.edu).

B. Li 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).

Manuscript received October 23, 2018.

Current network operators mainly rely on preset threshold systems for anomaly detection while the subsequent identification and reasoning procedures are conducted by experienced technicians. However, owing to the heterogeneity (heterogeneity in vendor devices, optical transmission technologies, applications [8]–[10], etc.), uncertainty (uncertainty in device conditions, alien wavelength configurations [11], etc.) and dynamicity of optical networks [12], [13], developing network-wide threshold systems that are effective over time is difficult. Loose thresholds may lead to low detection rates or prolonged detection delays, whereas tight thresholds can trigger vast false alarms, overburdening the control and management systems. Hence, excessive resource redundancies are usually reserved to provide guaranteed performance margins against potential anomalies [14]. In the meantime, manual anomaly identification and reasoning operations are laborious and they frustrate rapid evolutions of optical networks.

Recently, machine learning (ML) has shown appealing prospect of facilitating enhanced resource efficiency, QoS assurances and scalability in optical networks [15]. Specifically, ML enables network operators to realize knowledge-based autonomous service provisioning [16] by modeling complicated network behaviors (e.g., end-to-end quality-of-transmission [11], [17], traffic profile [18]–[20], etc.) and learning correct online provisioning policies from dynamic network operations [21], which are intractable with conventional theoretical approaches. The application of ML for fault management in optical networks has attracted extensive research attention lately [22]–[26]. In [22], Vela *et al.* analyzed four types of soft failures affecting the bit-error-rate (BER) of lightpaths and proposed two finite state machine based algorithms for detecting significant BER changes and identifying the corresponding failures patterns. Taking into account similar failure scenarios, the same authors also investigated ML-aided algorithms for soft failure localization [23]. The authors of [24] compared the performance of different ML algorithms in terms of complexity and accuracy for anomaly detection and identification in optical networks. In [25], Rafique *et al.* proposed a cognitive assurance architecture on the basis of software-defined networking (SDN) and developed a neural network based classifier to assist anomaly detection. Different from previous works that focus on exploiting the characteristic of BER changes, the authors of [25] evaluated their proposal with experimental data from optical power measurement un-

der diverse failure modes. In [26], an ML-aided framework, together with two classification algorithms, were proposed for detecting and identifying optical network jamming signal attacks of varying intensities.

The above ML-based anomaly detection and identification schemes suffer from a fundamental issue: they employ supervised learning models relying on specific knowledge of abnormal network behaviors and large amount of anomaly data which are difficult to obtain in a real network scenario where anomalies occur infrequently. On the other hand, it is known that anomalies typically exhibit unique patterns deviating from normal network behaviors [22], [27]. Unsupervised ML techniques that can learn patterns of data by directly analyzing the similarities among data instances thereby would become promising tools for identifying anomalies from huge volumes of monitoring data.

In this context, our previous work [28] proposed a self-taught anomaly detection framework with a hybrid unsupervised and supervised ML approach. The proposed framework employs an unsupervised data clustering module (DCM) to analyze the patterns of optical performance monitoring data. Then, a supervised data regression and classification module (DRCM) is trained with the learned patterns for online anomaly detection. Such a self-learning/-taught mechanism potentially enables detecting unforeseen anomaly without requiring prior knowledge of abnormal network behaviors. This paper extends the conference paper [28] by providing a more comprehensive description of the self-taught anomaly detection framework and the designs of the DCM and DRCM, and presenting a new set of results considering two use cases. In particular, we detailed the DCM design with the density-based clustering algorithm and elaborated on the principle of the DRCM enabled by a deep neural network (DNN) architecture. We evaluated the performance of the proposed framework with experimental data of both single-point and end-to-end detections. Results show that below 1% false positive and false negative rates can be achieved.

The organization of the paper is as follows. Section II presents the proposed self-taught anomaly detection framework. Sections III and IV detail the designs of the DCM and DRCM, respectively. Section V provides the evaluation results and related discussions. Finally, Section VI summarizes the paper and discusses potential future research topics.

II. Self-taught Anomaly Detection Framework

Fig. 1 shows the schematic of the proposed self-taught framework for anomaly detection in optical networks. The framework requires that network operators deploy optical performance monitoring (OPM) modules at certain network locations to perform real-time surveillance of data plane operations, e.g., monitoring the spectrum utilization, signal power, noise level and so forth, on different links. SDN-based network telemetry services [29] can be utilized to assist remote and on-demand monitoring data collection. The data preprocessing module (DPM) retrieves the obtained performance monitoring data from the database and tailors the original data (i.e., feature engineering) for different anomaly detection purposes. For instance, to surveil the behavior of an end-to-end lightpath, DPM would generate a new proprietary dataset by extracting and concatenating the lightpath's parameters at different monitoring sites over time.

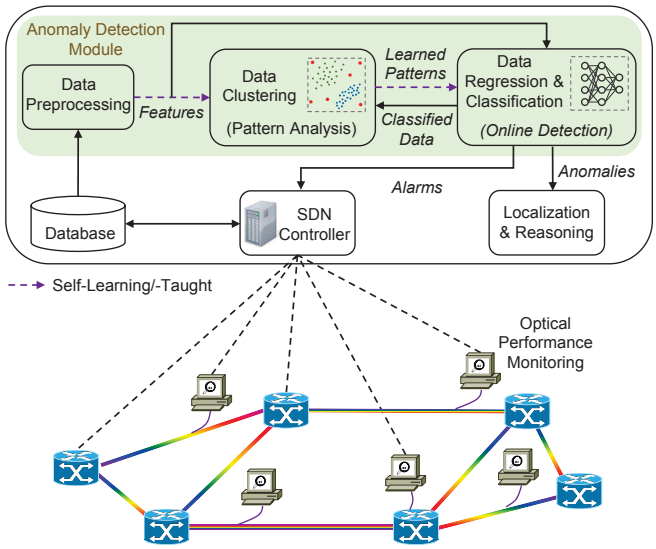

Fig. 1. Proposed self-taught anomaly detection framework.

The learning phase applies a hybrid unsupervised and supervised learning approach. Firstly, the processed data are input to an unsupervised DCM for pattern analysis (a *self-learning* mechanism). The DCM exploits the similarities among data instances and hereby divides the performance data into a number of clusters and outliers. Following a consensual assumption that the occurrence of network anomalies are much less frequent than that of normal behaviors, the DCM labels outliers as anomalies. Since the DCM learns the patterns of the input data directly without relying on any prior knowledge about abnormal behaviors, it can potentially detect unknown anomalies in optical networks. Then, the learned patterns are transferred to train a supervised DRCM for online anomaly detection (a *self-taught* mechanism). Specifically, the DRCM examines each new data instance by integrating the functions of both a regressor that predicts key features of the instance, and a classifier that attempts to classify the instance into one of the classes identified by the DCM. The rationale of employing a DRCM for online detection is that once trained, its time complexity is fixed, mainly determined by the scale of the adopted supervised learning model (i.e., the number of model parameters). On the contrary, the complexity of the DCM scales up with the size of the database, and can become an issue as it has to traverse the whole database every time a new data instance is received.

During online operations, the DRCM raises an alarm upon detecting an anomaly and inform the SDN controller, which in turn can take certain reactions (e.g., service reconfigurations) to mitigate the risk of potential severe service disruptions. The DRCM also sends out the abnormal data instance for further anomaly localization and reasoning. In the meantime, the hybrid learning process is periodically invoked for attaining the state-of-art network behaviors.

III. Design of the DCM

The distribution of optical performance monitoring data may present irregular shapes. Fig. 2 shows such an example in a two-dimensional space. In this context, we designed the DCM with the density-based clustering algorithm in [30], which is known to be able to identify any shape of clusters.

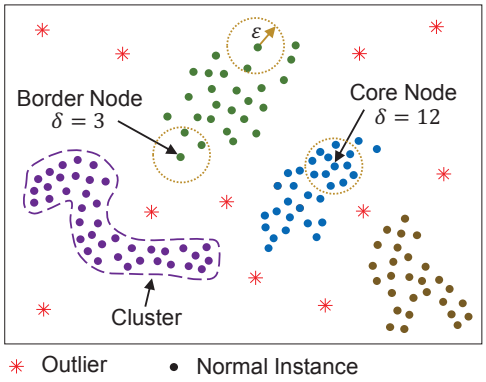

Fig. 2. An example showing the principle of density-based clustering.

Let S denote the input dataset and $d_{i,j}$ represent the distance between data instances s_i and s_j ($s_i, s_j \in S$). In this work, we measure $d_{i,j}$ with the Euclidean distance as it has been widely applied in the anomaly detection domain thanks to its capability of showing clear differences between normal and abnormal instances [31]. In particular, we calculate $d_{i,j}$ as,

$$d_{i,j} = \sqrt{\sum_{k}(s_{i,k} - s_{j,k})^2}, \tag{1}$$

where $s_{i,k}$ is the k-th dimension of s_i. The following definitions are prerequisites for discussing the density-based clustering algorithm. Firstly, the ε-neighborhood of an instance s_i is defined as the set of instances whose distances to s_i are within ε, i.e.,

$$E_i = \{s_j | d_{i,j} \leq \varepsilon, \forall s_j\}. \tag{2}$$

The dashed circles in Fig. 2 show the ε-neighborhoods of the corresponding centric nodes[1]. We refer to the size of E_i as the density of s_i, denoted as δ_i. Then, the core node condition for each s_i is defined as,

$$\delta_i \geq MinPts, \tag{3}$$

where $MinPts$ is a preset parameter for the algorithm. Lastly, s_j is called directly density-reachable from s_i if $s_j \in E_i$ and $\delta_i \geq MinPts$. The basic idea of density-based clustering stems from the intuition that clusters normally center on instances with high densities. Algorithm 1 summarizes the principle of density-based clustering [30]. In each iteration, the algorithm starts from a core node and iteratively add density-reachable instances from it to form a cluster. Specifically, in line 2, the distance between each pair of instances is first calculated. The for-loop covering lines 3-22 goes through every instance in the dataset and expands from the instance to form a new cluster if the instance has not be clustered and satisfies the core node condition. The inner loop from lines 9-20 accomplishes the recursive expansion. We initiate a new cluster if the ε-neighborhoods of all the core nodes in the current cluster have been included (line 9). Finally, all the instances that cannot be clustered into any of the identified clusters are categorized as outliers/anomalies.

[1]We use "node" and "data instance" interchangeably in the rest of the paper.

Algorithm 1: Procedures of density-based clustering.

Input: Dataset S, ε, $MinPts$
Output: Set of clusters C, set of outliers U

1 $C = \emptyset$;
2 calculate $d_{i,j}, \forall s_i, s_j \in S$;
3 **for** *each* $s_i \in S$ **do**
4 **if** s_i *belongs to any cluster OR* $\delta_i < MinPts$ **then**
5 **continue**;
6 **end**
7 $\hat{C} = \{s_i\}$, $\Lambda = E_i$;
8 remove from Λ the instances belonging to C or $\hat{C}$;
9 **while** $\Lambda \neq \emptyset$ **do**
10 store Λ in $\hat{C}$;
11 $\Gamma = \emptyset$;
12 **for** *each* $s_j \in \Lambda$ **do**
13 **if** $\delta_j \geq MinPts$ **then**
14 $\hat{\Gamma} = E_j$;
15 remove from $\hat{\Gamma}$ the instances belonging to C, $\hat{C}$, or Γ;
16 store $\hat{\Gamma}$ in Γ;
17 **end**
18 **end**
19 $\Lambda = \Gamma$;
20 **end**
21 store $\hat{C}$ in C;
22 **end**
23 store instances that do not belong to any cluster in U;

We developed a simple method to facilitate determining proper values for ε and $MinPts$. First, based on the assumption that the number of anomalies is much smaller than that of normal instances, we can set $MinPts$ as a small number, i.e., assuming that there will not be $MinPts$ simultaneous anomalies of the same types. Typically, we can set $MinPts = 4$ according to [30]. For larger-scale datasets with sufficient amounts of normal data, a larger value of $MinPts$ may be applied for improved anomaly detection rate. Then, we can determine the value of ε by gradually increasing ε from a very small value and observing the variation in the number of detected anomalies, i.e., $|U|$. In the beginning, $|U|$ decreases sharply because a larger ε encourages forming of clusters. As the true anomalies are located farther away from neighboring nodes compared with normal instances, the decreasing rate of $|U|$ will become very low at a certain point (say ε^*) when the majority of normal instances have been clustered. In other words, a much larger increase of ε is required for the algorithm to further include anomalies in normal clusters. Therefore, we can set ε as ε^*.

The complexity of Algorithm 1 is $O(|S|^2)$. Note that, when applying the density-based clustering algorithm to online anomaly detection, we need to revisit every instance in the

dataset (in the worst case) for each newly collected instance to check whether it can be included in one of the identified clusters. Such operations introduce a complexity of $O(|S|)$. Moreover, there should also be a procedure to readjust (e.g., merge or further expand) the identified clusters upon the joining of a new instance. As the complexity of the DCM scales up with $|S|$, it may become a bottleneck to online operations. Restricting $|S|$ by discarding some non-critical [2] or outdated data instances can be beneficial but does not resolve the problem essentially. Therefore, a time-efficient scheme irrelevant of $|S|$ is desired for realizing online anomaly detection.

IV. Design of the DRCM

Recall the example in Fig. 2, one intuitive observation is that the patterns of data, i.e., the sizes, shapes and locations of clusters, critically determine the identifications of new data instances. Further, if we can teach a module to acquire such knowledge learned by the DCM, we can thereby omit the traversal of the dataset during each online detection and significantly reduce the complexity. To this end, we first encode the output of the DCM by constructing a new labeled dataset $S' = \{(s_i, l_i), s_i \in S\}$, where each original instance s_i is labeled with its cluster ID l_i ($l_i \in \{1, ..., |C|+1\}$). Here, all the abnormal instances are assigned cluster ID $|C| + 1$. Then, we can train a classifier with S' and make it predict the cluster ID of a new instance. However, the amount of abnormal data is usually small and cannot represent the space of anomalies very well. Applying the classifier alone would result in low detection accuracy. For instance, an anomaly may be incorrectly classified into one of the surrounding clusters if it is close to the cluster boundaries or no similar anomaly has been detected previously. On the other hand, the discussions in Section III have suggested that node densities actually provide more comprehensive information about the distribution of the data than the final clustering results. Typically, border nodes of clusters have lower densities than core nodes while the densities of anomalies are the lowest, yielding a potential field of densities. This gives us a hint that we can leverage such an additional dimension of knowledge and train a density regressor to work cooperatively with the classifier, i.e., employing the DRCM, for more accurate anomaly detection. Specifically, the DRCM predicts the density $\tilde{\delta}_i$ (*regression*) and cluster ID $\tilde{l}_i$ (*classification*) of each new instance $\tilde{s}_i$ simultaneously. $\tilde{s}_i$ is detected as an anomaly when either $\tilde{l}_i = |C|+1$ or $\tilde{l}_i \neq |C|+1$ but $\tilde{\delta}_i$ is smaller than a preset threshold $\delta_0(\tilde{l}_i)$ which should be determined by the minimum density value of instances in cluster $\tilde{l}_i$. Here, we call the latter case as weakly positive as by definition, the density of a border node can be as low as those of anomalies. For such case, we invoke an additional validation process by comparing $\tilde{s}_i$ with instances in cluster $\tilde{l}_i$ according to the principle of density-based clustering. Note that, node densities can range from 1 to tens of thousands, but the DRCM is only sensitive to prediction errors on densities of small values. For instance, mistakenly predicting $\delta_i = 1$ as $\tilde{\delta}_i = 5$ may completely change the judgement of the DRCM (failed to detect an anomaly), whereas an error of $\delta_i = 1000$ to $\tilde{\delta}_i = 1100$ would not affect the performance of the DRCM. Hence, we make the DRCM predict the logarithms instead of the absolute values of densities, concentrating on especially optimizing the prediction accuracy regarding low-density instances.

[2]Non-critical instances refer to instances whose removals do not change the structures of the identified clusters.

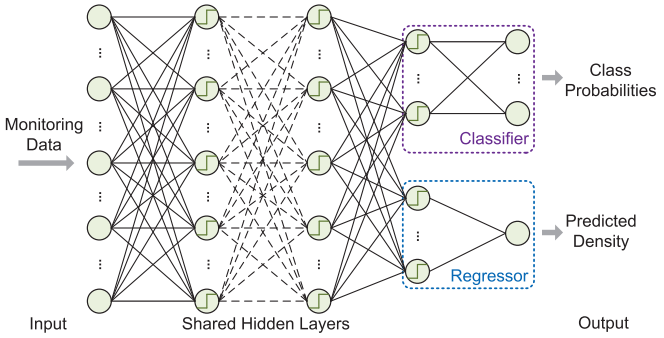

Fig. 3. Architecture of the DNN-based DRCM design.

We designed the DRCM with the DNN architecture due to its well-recognized capability of representing high-dimensional data and molding complex functions. Instead of implementing the regressor and the classifier with two separate DNNs, we integrated them into one unified DNN structure for lower system complexity and better scalability. Fig. 3 shows the detailed structure of the DRCM. The DRCM takes as input each monitoring data instance s_i. The input is processed by a few shared fully-connected hidden layers for feature extraction. Each neuron $v_{n,m}$ in hidden layer n calculates its output as,

$$h_{n,m} = g\left(\left(w_{n-1}^{n,m}\right)^T h_{n-1} + b_{n,m}\right), \tag{4}$$

where $g(\cdot)$ represents the activation function, $w_{n-1}^{n,m}$ is the vector containing the weights of connections from neurons in layer $n-1$ to $v_{n,m}$, and $b_{n,m}$ is the bias. Two separate blocks are deployed after the shared hidden layers for the regression and classification tasks, respectively. Each of the blocks is a neural network of two layers. The regressor outputs the real-valued predicted density $\tilde{\delta}_i$ with no activation function for the output layer. The classifier applies the *Softmax* function to generate the class probabilities p_i^c ($c \in \{1, ..., |C| + 1\}$) to assist classification[3]. In particular, the *Softmax* function is given by,

$$g(x, c) = \frac{e^{x_c}}{\sum_t e^{x_t}}. \tag{5}$$

We define the regression and classification losses as the mean squared error and the cross-entropy loss, respectively, i.e.,

$$L_{re}(\theta) = \frac{1}{|S|} \sum_{s_i \in S} \left(\delta_i - \tilde{\delta}_i\right)^2, \tag{6}$$

$$L_{cl}(\theta) = -\frac{1}{|S|} \sum_{s_i \in S} \sum_c \left(\chi_i^c \log p_i^c + (1 - \chi_i^c) \log\left(1 - p_i^c\right)\right), \tag{7}$$

[3]Note that, during online operations when $|C|$ can potentially increase with the growth of S, we may need to extend the scale of the classifier appropriately upon periodical system retraining. Alternatively, we can obviate changing the classifier architecture by combining normal classes to fix the number of classes as 2 (i.e., normal and abnormal).

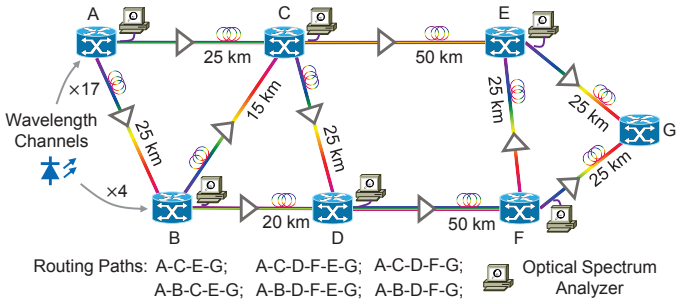

Fig. 4. Testbed setup for dataset generation.

where θ is the set of parameters of the DNN, χ_i^c equates to 1 if $l_i = c$ and 0 otherwise. Finally, the DNN can be trained (i.e., tuning θ) by minimizing the overall loss as,

$$L(\theta) = L_{re}(\theta) + \gamma L_{cl}(\theta) + \varsigma \left|\left|\theta\right|\right|^2, \tag{8}$$

where γ and ς are weighting coefficients and $\left|\left|\theta\right|\right|^2$ is the regularization loss introduced to prevent overfitting.

After detecting an anomaly with the DRCM, we can perform anomaly localization and reasoning by first examining whether similar anomalies have been identified. If not, we proceed to compare the detected anomaly with normal instances surrounding it. In particular, we can calculate the distance between two instances in each dimension and possibly discern from the distance vector (1) distinct variations in certain dimensions, e.g., an abrupt increase of the noise level at an intermediate node of a lightpath caused by the amplifier malfunction, or (2) a unique pattern, e.g., a gradual decrease of the channel power gain due to the amplifier aging. Meanwhile, if multiple concurrent anomalies are detected, correlation schemes [32] can be applied for localizing the faults.

V. Performance Evaluation

A. Dataset Generation

We evaluated the performance of the proposed self-taught anomaly detection framework by using experimental data collected from a 7-node optical network testbed (see Fig. 4) with six different routing paths. In particular, we launched 17 and four wavelength channels at Nodes A and B, respectively. By reconfiguring the wavelength selective switches (WSSs) parameters (i.e., ports' bandwidth and attenuation) at each node, we created diverse network link load scenarios. For each scenario, we processed the readings from the optical spectrum analyzers (OSAs) located at six fixed locations to obtain the values of optical power at each wavelength and out-of-band noise floor, which constitute the original dataset. For a small portion of these network instances, we purposely increased the attenuations introduced by the WSSs for certain specific wavelength channels, emulating abnormal power distributions in certain links. Fig. 5 shows an example of normal and abnormal data instances from the experiment. In this way, we emulated soft-failures such as the malfunctioning of a tunable filter [22], a power equalizer or an erbium-doped fiber amplifier (EDFA), or a jamming attack [4], which can also cause abnormal power distributions.

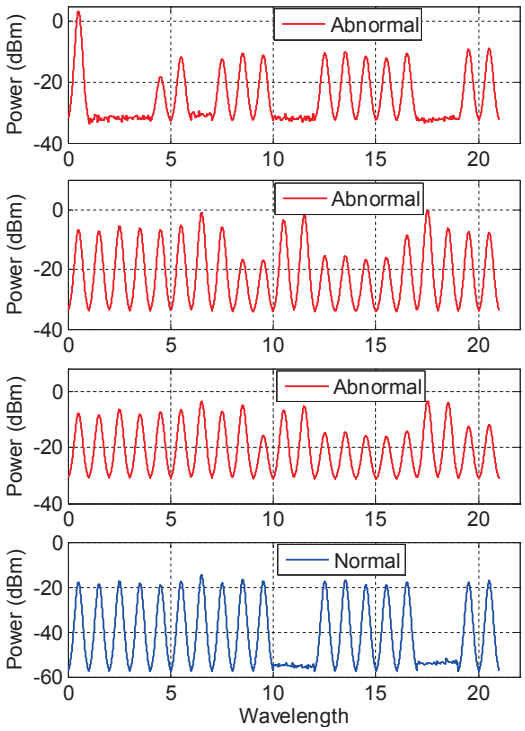

Fig. 5. Examples of normal and abnormal data instances.

B. Use Case I: Single-Point Anomaly Detection

We focused on detecting abnormal patterns of data collected at each monitoring point. Each data instance contains 22 features, including the power of 21 wavelength channels (i.e., the 17 and four wavelength channels launched at Nodes A and B) and the noise floor. In total, we obtained $18,680$ normal instances and 50 anomalies. All the data instances were scaled [33] before being processed by the DCM and DRCM. We measured distances among data instances as their Euclidean distances (see Eq. 1).

To determine the correct configuration of ε for the DCM, we first set $MinPts = 4$ and plot the variation in the number of detected anomalies as function of ε in Fig. 6. Based on the parameter selection method presented in Section III, we can see that ε should be assigned a value of around 0.4. Table I shows the results of false negative rate (denoted as f_n) and false positive rate (denoted as f_p) from the DCM with different setup of $MinPts$ and ε. The results indicate that a larger ε leads to higher f_n but lower f_p, whereas a larger $MinPts$ results in an opposite trend. This is because with larger values of ε (when $MinPts$ is fixed), anomalies are more likely to be included in the ε-neighborhoods of normal instances, making them more difficult to be detected. On the contrary, increasing $MinPts$ discourages the forming of clusters and generates more outliers, which facilitates anomaly detection but raises more false alarms as well. With $MinPts$ and ε being set as 4 and 0.4, respectively, the DCM can achieve up to 100% anomaly detection and 0.01% false positive rate for the dataset under evaluation, which clearly demonstrates the effectiveness of the proposed method.

We compared the performance of the DCM with that of the K-means clustering algorithm [34]. Table II presents the results

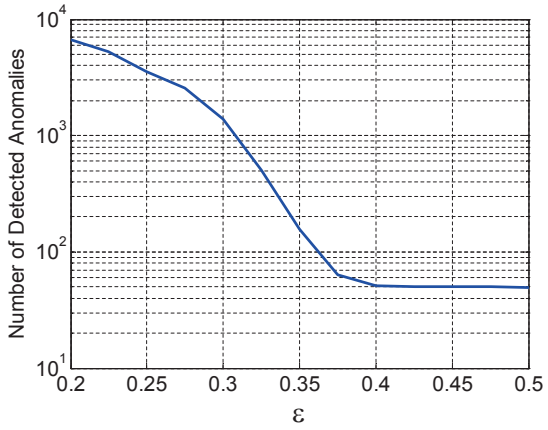

Fig. 6. Variation in the number of detected anomalies as function of ε.

TABLE I
FALSE NEGATIVE AND FALSE POSITIVE RATES OF THE DCM ((f_n, f_p) %).

ε \ $MinPts$	2	4	6
0.20	0.0, 30.3	0.0, 35.6	0.0, 48.0
0.30	0.0, 4.12	0.0, 7.17	0.0, 22.9
0.35	0.0, 0.30	0.0, 0.57	0.0, 12.0
0.40	4.0, 0.0	0.0, 0.01	0.0, 7.19
0.45	8.0, 0.0	***0.0, 0.0***	0.0, 4.80
0.50	1.00, 0.0	2.00, 0.0	2.00, 4.00

of f_n and f_p from *K*-means. Since *K*-means does not define outliers and groups all the instances into clusters, we slightly modified *K*-means to make it detect instances that belong to a certain ratio (η) of the farthest nodes to the core node of each cluster as anomalies. Meanwhile, *K*-means requires that the number of clusters N is provided as an input parameter. We can see that the lowest false negative rate *K*-means can achieve is 28.0%, which is inadequate for anomaly detection. The reason is that the application of *K*-means is only limited to problems with spherical clusters other than those with clusters of irregular shapes (see the example in Fig. 2).

According to the clustering result of the DCM (with $MinPts = 4$ and $\varepsilon = 0.4$), we obtained 621 normal classes and 1 abnormal class, as well as the density of each data instance. We implemented the DRCM with a DNN consisting of four shared hidden layers ($[32, 32, 32, 32]$), a classification block of two layers ($[32, 622]$) and a regression block of two layers ($[16, 1]$). Except for the output layers, we used *ELU* as the activation function. For the loss function in Eq. 8, γ and ς were set as 0.1 and 10^{-4}, respectively[4]. We conducted 10 independent experiments by randomly dividing the dataset into the training, validation and testing sets with a ratio of $7 : 1 : 2$. Fig. 7 shows the training and validation losses of the DRCM averaged from 10 experiments, indicating that the training converges without notable overfitting. Evaluations with the testing set show that the DRCM can achieve a mean square error of 14.4 for density prediction and an average

[4]We have tested different DNN architectures and parameters for the DRCM and this paper presents the most appropriate setup.

TABLE II
FALSE NEGATIVE AND FALSE POSITIVE RATES OF *K*-MEANS ((f_n, f_p) %).

N \ η	0.03	0.05	0.08
10	70.0, 2.90	70.0, 4.91	64.0, 7.90
20	44.0, 2.80	38.0, 4.81	***28.0, 7.77***
30	44.0, 2.78	34.0, 4.78	32.0, 7.76
40	50.0, 2.78	46.0, 4.79	28.0, 7.72

classification accuracy of 94.8%. Table III presents the results of the DRCM for anomaly detection. Recall that we defined a preset threshold to assist triggering validation processes for cases where instances are classified into normal classes but their predicted densities are lower than the threshold. For the sake of simplicity, a unified threshold δ_0 was applied for all the classes. By setting δ_0 as 0, we made the DRCM only rely on the classification results. Here, we denote the frequency of validation as f_{val}, which is a performance metric defined as the ratio of the number of instances triggering the validation process to the total number of instances under examination. We can observe that the DRCM fails to detect more than $1/3$ of the anomalies without taking into account the predicted densities. This is because the number of anomaly samples in the training set is too small to provide sufficient supervision signals for training a DNN classifier that can accurately detect anomalies. By introducing δ_0, we remarkably improved the anomaly detection rate of the DRCM but at the cost of increased f_{val}, which shows a trade-off between detection accuracy and system complexity. The false negative rate could be reduced to below 1% with f_{val} being 29.82%.

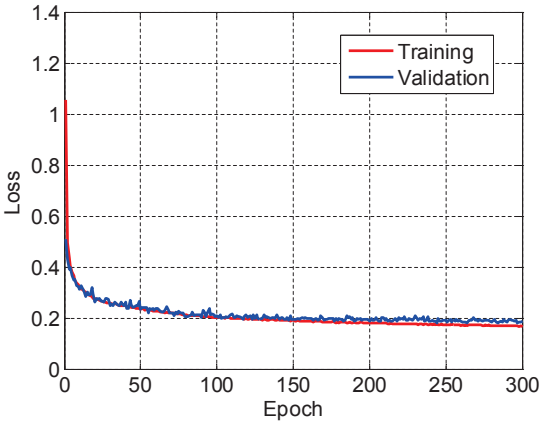

Fig. 7. Losses of the DRCM during training.

TABLE III
FALSE NEGATIVE AND FALSE POSITIVE RATES, AND FREQUENCY OF VALIDATIONS OF THE DRCM (%).

δ_0	0	3	5	7	9
f_n	36.36	24.54	10.91	1.82	0.91
f_p	0.07	0.07	0.07	0.07	0.07
f_{val}	0.0	0.60	10.24	22.81	29.82

C. Use Case II: End-to-End Anomaly Detection

Next, we conducted anomaly detection with performance data of end-to-end lightpaths. In particular, we focused on analyzing the behavior of a specific lightpath (i.e., lightpath A-B-D-F-E-G at wavelength 7) and obtained 300 normal and 10 abnormal data instances by extracting and combining the signal power and out-of-band noise floor at each monitoring point along the lightpath. Again, we scaled the data and measured the Euclidean distances. We did not apply the parameter selection method for this use case as the dataset is too small. Table IV summarizes the false positive and false negative rates of the DCM with different setup of ε and $MinPts$. Basically, the results in Table IV show a similar trend with the ones from Table I. In the best cases, the DCM can achieve 100% anomaly detection without any false alarm. Note that, in real network operations where normal and abnormal scenarios are more diverse and where larger volumes of data are available, the parameters and performance of the proposed approach may differ from those in this experiment. Meanwhile, we do not show the results from the DRCM because the amount of data is too small to train a DNN.

TABLE IV
FALSE NEGATIVE AND FALSE POSITIVE RATES OF THE DCM FOR END-TO-END DETECTION ((f_n, f_p) %).

ε \ $MinPts$	2	3	4
0.8	***0.0, 0.0***	***0.0, 0.0***	0.0, 18.97
1.0	***0.0, 0.0***	***0.0, 0.0***	0.0, 16.65
1.2	10.0, 0.0	10.0, 0.0	10.0, 12.26
1.4	50.0, 0.0	10.0, 0.0	10.0, 8.33

VI. CONCLUSION

In this paper, we proposed a self-taught anomaly detection framework based on a hybrid unsupervised and supervised ML approach. The proposed framework employs an unsupervised DCM for pattern analysis and a supervised DRCM for online anomaly detection. We designed the DCM and DRCM based on the density-based clustering algorithm and the DNN structure, respectively. Assessments based on experimental data demonstrated below 1% false positive and false negative rates when using the proposed framework.

Our future work will aim at improving the performance of the DRCM by collecting more normal data, taking into account more comprehensive and diverse failure scenarios, and applying more effective DNN architectures and training schemes. Other potential research topics include: (1) extending the current framework to incorporate cognitive anomaly localization and reasoning functions and (2) developing anomaly detection frameworks for multi-domain multi-layer networks with hierarchical monitoring [35] and multi-agent ML approaches [36].

ACKNOWLEDGMENTS

This work was supported in part by DOE DE-SC0016700, and NSF ICE-T:RC 1836921. The authors would like to thank G. Liu and K. Zhang at UC Davis for providing the experimental dataset. The authors would also like to thank Prof. Luis Velasco from Universitat Politécnica de Catalunya (UPC) for the fruitful discussions and helpful suggestions on anomaly detection scenarios in optical networks.

REFERENCES

[1] J. Zhang and B. Mukheriee, "A review of fault management in WDM mesh networks: basic concepts and research challenges," *IEEE Netw.*, vol. 18, no. 2, pp. 41–48, Mar. 2004.
[2] P. Lu, L. Zhang, X. Liu, J. Yao, and Z. Zhu, "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.
[3] C. Mas, I. Tomkos, and O. Tonguz, "Failure location algorithm for transparent optical networks," *IEEE J. Sel. Areas Commun.*, vol. 23, no. 8, pp. 1508–1519, Aug. 2005.
[4] R. Rejeb, M. Leeson, and R. Green, "Fault and attack management in all-optical networks," *IEEE Commun. Mag.*, vol. 44, no. 11, pp. 79–86, Nov. 2006.
[5] X. Chen, M. Tornatore, S. Zhu, F. Ji, W. Zhou, C. Chen, D. Hu, L. Jiang, and Z. Zhu, "Flexible availability-aware differentiated protection in software-defined elastic optical networks," *J. Lightwave Technol.*, vol. 33, no. 18, pp. 3872–3882, Sep.
[6] X. Chen, S. Zhu, L. Jiang, and Z. Zhu, "On spectrum efficient failure-independent path protection p-cycle design in elastic optical networks," *J. Lightw. Technol.*, vol. 33, no. 17, pp. 3719–3729, Sep. 2015.
[7] Z. Cheng, X. Zhang, S. Shen, S. Yu, J. Ren, and R. Lin, "T-trail: Link failure monitoring in software-defined optical networks," *J. Opt. Commun. Netw.*, vol. 10, no. 4, pp. 344–352, Apr. 2018.
[8] L. Gong, X. Zhou, X. Liu, W. Zhao, W. Lu, and Z. Zhu, "Efficient resource allocation for all-optical multicasting over spectrum-sliced elastic optical networks," *J. Opt. Commun. Netw.*, vol. 5, pp. 836–847, Aug. 2013.
[9] L. Gong and Z. Zhu, "Virtual optical network embedding (VONE) over elastic optical networks," *J. Lightw. Technol.*, vol. 32, pp. 450–460, Feb. 2014.
[10] 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*, Apr. 2014, pp. 1–9.
[11] R. Proietti, X. Chen, K. Zhang, G. Liu, M. Shamsabardeh, A. Castro, L. Velasco, Z. Zhu, and S. J. B. Yoo, "Experimental demonstration of machine-learning-aided QoT estimation in multi-domain elastic optical networks with alien wavelengths," *J. Opt. Commun. Netw.*, vol. 11, no. 1, pp. A1–A10, Jan. 2019.
[12] 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.
[13] M. Zhang, C. You, H. Jiang, and Z. Zhu, "Dynamic and adaptive bandwidth defragmentation in spectrum-sliced elastic optical networks with time-varying traffic," *J. Lightw. Technol.*, vol. 32, pp. 1014–1023, Mar. 2014.
[14] Y. Pointurier, "Design of low-margin optical networks," *J. Opt. Commun. Netw.*, vol. 9, no. 1, pp. A9–A17, 2017.
[15] F. Musumeci, C. Rottondi, A. Nag, T. Macaluso, D. Zibar, M. Ruffini, and M. Tornatore, "A survey on application of machine learning techniques in optical networks," *arXiv preprint arXiv:1803.07976*, 2018.
[16] X. Chen, R. Proietti, H. Lu, A. Castro, and S. J. B. Yoo, "Knowledge-based autonomous service provisioning in multi-domain elastic optical networks," *IEEE Commun. Mag.*, vol. 56, no. 8, pp. 152–158, Aug. 2018.
[17] S. Oda, M. Miyabe, S. Yoshida, T. Katagiri, Y. Aoki, T. Hoshida, T. Rasmussen, M. Birk, and K. Tse, "A learning living network with open ROADMs," *J. Lightw. Techn.*, vol. 35, no. 8, pp. 1350–1356, Apr. 2017.
[18] X. Chen, J. Guo, Z. Zhu, A. Castro, R. Proietti, H. Lu, M. Shamsabardeh, and S. J. B. Yoo, "Leveraging deep learning to achieve knowledge-based autonomous service provisioning in broker-based multi-domain SD-EONs with proactive and intelligent predictions of multi-domain traffic," in *Proc. of ECOC*, Sept. 2017, pp. 1–3.
[19] 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.

[20] J. Guo and Z. Zhu, "When deep learning meets inter-datacenter optical network management: Advantages and vulnerabilities," *J. Lightw. Technol.*, vol. 36, no. 20, pp. 4761–4773, Oct 2018.
[21] X. Chen, J. Guo, Z. Zhu, R. Proietti, A. Castro, and S. J. B. Yoo, "Deep-RMSA: A deep-reinforcement-learning routing, modulation and spectrum assignment agent for elastic optical networks," in *Proc. of OFC*, Mar. 2018, pp. 1–3.
[22] A. Vela, M. Ruiz, F. Fresi, N. Sambo, F. Cugini, G. Meloni, L. Potí, L. Velasco, and P. Castoldi, "BER degradation detection and failure identification in elastic optical networks," *J. Lightw. Technol.*, vol. 35, no. 21, pp. 4595–4604, Nov. 2017.
[23] A. Vela, B. Shariati, M. Ruiz, F. Cugini, A. Castro, H. Lu, R. Proietti, J. Comellas, P. Castoldi, S. J. B. Yoo, and L. Velasco, "Soft failure localization during commissioning testing and lightpath operation," *J. Opt. Commun. Netw.*, vol. 10, no. 1, pp. A27–A36, Jan. 2018.
[24] S. Shahkarami, F. Musumeci, F. Cugini, and M. Tornatore, "Machine-learning-based soft-failure detection and identification in optical networks," in *Proc. of OFC*, Mar. 2018, pp. 1–3.
[25] D. Rafique, T. Szyrkowiec, H. Grießer, A. Autenrieth, and J. Elbers, "Cognitive assurance architecture for optical network fault management," *J. Lightw. Technol.*, vol. 36, no. 7, pp. 1443–1450, Apr. 2018.
[26] C. Natalino, M. Schiano, A. Giglio, L. Wosinska, and M. Furdek, "Field demonstration of machine-learning-aided detection and identification of jamming attacks in optical networks," in *Proc. of ECOC*, Sept. 2018, pp. 1–3.
[27] V. Chandola, A. Banerjee, and V. Kumar, "Anomaly detection for discrete sequences: A survey," *IEEE Trans. Knowl. Data Engi.*, vol. 24, no. 5, pp. 823–839, May 2012.
[28] X. Chen, B. Li, M. Shamsabardeh, R. Proietti, Z. Zhu, and S. J. B. Yoo, "On real-time and self-taught anomaly detection in optical networks using hybrid unsupervised/supervised learning," in *Proc. of ECOC*, Sept. 2018, pp. 1–3.
[29] F. Paolucci, A. Sgambelluri, F. Cugini, and P. Castoldi, "Network telemetry streaming services in SDN-based disaggregated optical networks," *J. Lightw. Technol.*, vol. 36, no. 15, pp. 3142–3149, Jan. 2018.
[30] 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*, Aug. 1996, pp. 226–231.
[31] D. Weller-Fahy, B. Borghetti, and A. Sodemann, "A survey of distance and similarity measures used within network intrusion anomaly detection," *IEEE Commu. Surv. Tutor.*, vol. 17, no. 1, pp. 70–91, Firstquarter 2015.
[32] T. Panayiotou, S. Chatzis, and G. Ellinas, "Leveraging statistical machine learning to address failure localization in optical networks," *J. Opt. Commun. Netw.*, vol. 10, no. 3, pp. 162–173, Mar. 2018.
[33] sklearn.preprocessing.scale. [Online]. Available: http://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.scale.html
[34] J. Hartigan and M. Wong, "Algorithm AS 136: A K-Means clustering algorithm," *J. Royal Stat. Soc. Ser. C Appl. Stat.*, vol. 28, no. 1, pp. 100–108, 1979.
[35] G. Liu, K. Zhang, X. Chen, H. Lu, J. Guo, J. Yin, R. Proietti, Z. Zhu, and S. J. B. Yoo, "The first testbed demonstration of cognitive end-to-end optical service provisioning with hierarchical learning across multiple autonomous systems," in *Proc. of OFC*, 2018, pp. 1–3.
[36] X. Chen, B. Li, R. Proietti, Z. Zhu, and S. J. B. Yoo, "Multi-agent deep reinforcement learning in cognitive inter-domain networking with multi-broker orchestration," in *Proc. of OFC*, 2019, pp. 1–3.

Xiaoliang Chen received his Ph.D. degree from the University of Science and Technology of China in 2016. He is currently a research scholar at the University of California, Davis (UC Davis). His research interests include optical networks, software-defined networking, and cognitive networking. He has published more than 50 papers on the journals and conferences of IEEE and OSA. He is an Associate Editor of Springer's Telecommunication Systems Journal, Wiley's Emerging Telecommunications Technologies Journal, and KSII Transactions on Internet and Information Systems, and a TPC member of IEEE ICNC 2018, 2019, and ICC 2018.

Baojia Li is currently pursing the M.S. degree in the University of Science and Technology of China. His research interest includes elastic optical networking and cognitive networking.

Roberto Proietti received the M.S. degree in telecommunications engineering from the University of Pisa, Pisa, Italy, in 2004, and the Ph.D. degree in electrical engineering from Scuola Superiore Sant Anna, Pisa, in 2009. He is a Project Scientist with the Next Generation Networking Systems Laboratory at UC Davis. His research interests include optical switching technologies and architectures for supercomputing and data center applications, high-spectral-efficiency coherent transmission systems, and elastic optical networking.

Zuqing Zhu received his Ph.D. degree from UC Davis in 2007. He is currently a full professor at the University of Science and Technology of China. Prior to that, he worked in the Service Provider Technology Group of Cisco Systems, San Jose, California. His research focuses on optical networks, and he received the Best Paper Awards from IEEE ICC 2013, IEEE GLOBECOM 2013, IEEE ICNC 2014, IEEE ICC 2015, and IEEE ONDM 2018.

S. J. Ben Yoo is a Distinguished Professor at UC Davis. His research at UC Davis includes 2D/3D photonic integration for future computing, cognitive networks, communication, imaging, and navigation systems, micro/nano systems integration, and the future Internet. Prior to joining UC Davis in 1999, he was a Senior Research Scientist at Bellcore, leading technical efforts in integrated photonics, optical networking, and systems integration. His research activities at Bellcore included the next-generation Internet, reconfigurable multiwavelength optical networks (MONET), wavelength interchanging cross connects, wavelength converters, vertical-cavity lasers, and high-speed modulators. He led the MONET testbed experimentation efforts, and participated in ATD/MONET systems integration and a number of standardization activities. Prior to joining Bellcore in 1991, he conducted research on nonlinear optical processes in quantum wells, a four-wave-mixing study of relaxation mechanisms in dye molecules, and ultrafast diffusion-driven photodetectors at Stanford University (BS 84', MS 86', PhD 91', Stanford University). Prof. Yoo is Fellow of IEEE, OSA, NIAC and a recipient of the DARPA Award for Sustained Excellence (1997), the Bellcore CEO Award (1998), the Mid-Career Research Faculty Award (2004 UC Davis), and the Senior Research Faculty Award (2011 UC Davis).