On Board-level Failure Localization in Optical Transport Networks Using Graph Neural Network

Yan Jiao*, Pin-Han Ho*, Xiangzhu Lu*, János Tapolcai†, and Limei Peng‡
*Department of Electrical and Computer Engineering, University of Waterloo, Waterloo, Canada
†Department of Telecommunications and Media Informatics, Budapest University of Technology and Economics, Hungary
‡School of Computer Science and Engineering, Kyungpook National University, Daegu, Republic of Korea
Email: {y42jiao,p4ho,x244lu}@uwaterloo.ca, tapolcai@tmit.bme.hu, auroraplm@knu.ac.kr

***Abstract*—This paper investigates a novel framework for board-level failure localization in the Optical Transport Networks (OTN), dubbed Board-Alarm Propagation Tree based Failure Localization (BAPT-FL). Foremost, a collection of functional graphs (FGs) is garnered by iteratively tagging each board in the network topology, serving as the ground of the proposed framework. Concretely, BAPT-FL is designed to build a range of BAPTs by correlating the tagged boards and alarms involved in the FGs, where each BAPT deems a failed board and its correlated alarms as the root and leaves, respectively. To evaluate the edge weights of potential BAPTs induced by FGs, a graph neural network (GNN) with the graph transformer operator is employed as an edge classifier, which characterizes each vertex/edge from diverse dimensions including time, traffic distribution, network topology, and board/alarm attributes. Subsequently, we frame an integer linear programming (ILP) problem to construct the best possible BAPT(s). Extensive case studies are conducted to showcase BAPT-FL's advantage over its counterparts in terms of the metrics assessing failed boards/root alarms. We also delve into its performance in volatile environmental variations such as diverse failure scenarios, network topologies, traffic distributions, and noise alarms.**

***Index Terms*—board-level failure localization, Optical Transport Networks (OTN), graph neural network (GNN), integer linear programming (ILP)**

I. Introduction

Internet backbones have undergone explosive growth in bandwidth requirement and geographical coverage, aiming at adapting to the multi-service and multi-tenant heterogeneous network environment. Optical Transport Networks (OTN) are formalized by ITU-T as a set of telecommunication protocols [1], allowing for multiplexing different services onto a single high-capacity wavelength to underpin the rapid advancement of contemporary Internet backbones. The OTN optical layers encompass the optical transport section (OTS), optical multiplex section (OMS), and optical channel (OCH) [2]. Specifically, each OCH (i.e. lightpath) traverses various types of device boards connected in series, such as the fiber segment, optical amplifier (OA), fiber interface unit (FIU), optical multiplexer (OM), optical demultiplexer (OD), optical transponder unit (OTU), etc. [3] The OTN control plane harnesses a comprehensive set of rigid alarming mechanisms at each board for monitoring system performance. As a result, each board is capable of generating alarms in response to any failure event perceived by its sensor. For example, if an OD board detects a failure in its upstream fiber segment, it will notify the network management system (NMS) via an alarm. Alternatively, an OTN board will report an alarm to the NMS if any exception degrades the received lightpath's quality of service.

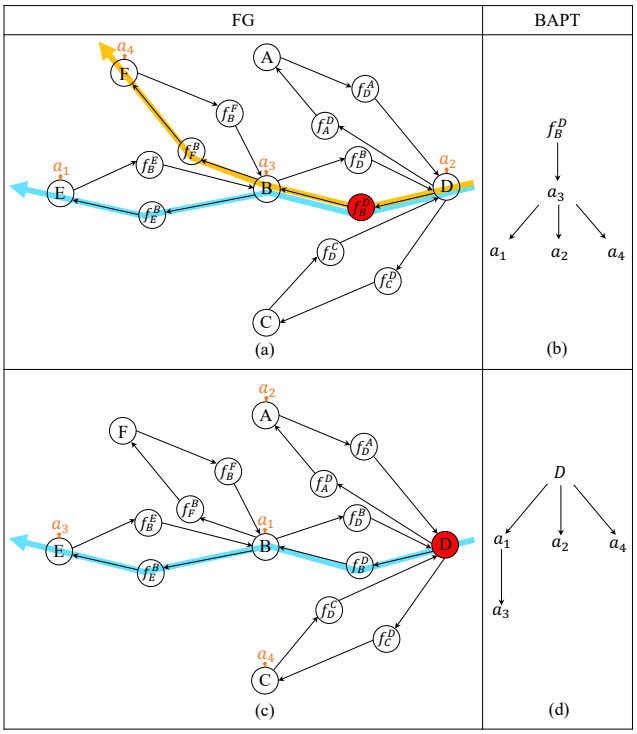


Fig. 1: (a), (c) Illustrations of two FGs due to single-board failure events, where $A, \cdots, F, f_D^A, \cdots, f_B^F$ are device boards, bold arrows are lightpaths, red nodes are failed tagged boards, and $a_1, \cdots, a_4$ are alarm types. (b), (d) Consequential BAPTs.

Overall, a failure event may interfere with one or multiple boards, resulting in tearing down numerous lightpaths by accident. Concurrently, other boards along the affected lightpaths may sense these disruptions as well. Additionally, a board might emit an alarm and send it to the NMS not only due to an identified failure event but also in response to the notification alarm stemming from another remote board. Accordingly, a failure event inevitably gives rise to an alarm flood, considerably boosting the complexity of alarm analysis and failure localization, particularly in the network domains with a myriad of boards such as the OTN.

Alarm correlation is universally acknowledged as an effective approach to discerning the dependencies among alarms. The objective is to remove those dependent alarms as many as possible, thereby substantially decreasing the complexity of failure localization. An illustrative example is given in Fig. 1(a). The fiber segment f_B^D is taken as the tagged board, attaining the *functional graph* (FG) that incorporates all boards traversed by the functional connections pertaining to the tagged board such as OTS, OMS, and OCH. A failure that hits f_B^D is first detected by its downstream board B. B issues an alarm a_3 and informs f_B^D's upstream board D via a_2, where these alarms are further reported to the NMS. In addition, the impacts of f_B^D are detected by all other boards traversing the affected lightpaths, engendering a_1 and a_4. Note that fiber segments are passive devices, thereby not reporting any alarms but could be failed.

Another example is illustrated in Fig. 1(c). The failure on the tagged board D even malfunctions its sensor and hence didn't raise any alarm. However, D's neighbouring boards A, B, and C receive the notification alarms a_2, a_1, and a_4, respectively. B causes E to report a_3, where E passes through the interrupted lightpath. Apparently, the failures on f_B^D and D influence an identical network topology of different traffic distributions, whereas they yield the same alarm pattern. Besides, the failure type and current network state determine whether an alarm is the root alarm or non-root alarm and its correlation with other alarms. For instance, in Fig. 1(a), a_3 is the root alarm and it triggers a_4. Conversely, in Fig. 1(c), a_4 is the root alarm, yet it didn't bring about a_3.

According to the examples above, the failure-alarm propagation phenomenon resulting from a failed tagged board in an FG is portrayed by a *Board-Alarm Propagation Tree* (BAPT), as displayed in Fig. 1(b) and Fig. 1(d). The goal is to recognize dependent alarms for precisely localizing failed device boards in the OTN optical layers. Extensive research achievements are directed toward proposing effective alarm correlation approaches to assist failure localization. Some of their findings concentrate on utilizing the relationships among alarms [5], [6], [9], [10], [13], [15]–[17], [25], [26]. Others exploit the relationships between failed nodes/links and alarms [7], [8], [11], [12], [14], [18]–[24]. Nevertheless, these works generally suppose that the network environment is simply static, which may not provide sufficient generality for the changeable network environment, especially for those experiencing alterations in network topology or traffic distribution.

Inspired by its importance and unique challenges, this paper introduces a novel board-level failure localization framework, termed *Board-Alarm Propagation Tree based Failure Localization* (BAPT-FL). Firstly, we assume that during an *observation window* (OW) of a given network state, the status of each board can change no more than once at the beginning of this OW, i.e., either maintaining normal or transitioning from normal to failed, incurring an alarm set. Then we define an FG as a tagged board-centric graph that contains all boards functionally related to the tagged one. Based on an FG, a BAPT represents the tree-shape correlation between the failed tagged board and alarms, where each board/alarm is featured by several attributes including its occurrence time and board type, and alarm type.

Given a set of FGs within an OW, BAPT-FL adopts the graph neural network (GNN) as an edge classifier for weighting the edges in each potential BAPT. The trained graph edge binary classifier jointly considers the network topology, traffic distribution, and received alarms. It is intended to accommodate diverse network environments, provided that they comply with the same failure-alarm propagation rules. With all weighted edges in place, *BAPT formation* can be exclusively realized by tackling an integer linear programming (ILP) problem, where each BAPT, connecting to one or multiple alarms, is rooted by a failed board. The purpose of BAPT formation is to cover all alarms with the minimum cost.

The contributions of this paper are summarized as follows.

- Propose a novel board-level failure localization framework, referred to as BAPT-FL, which leans on a graph edge binary classifier and BAPT modelling approach.
- Leverage GNN for building a novel graph edge binary classifier. This classifier offers superior adaptability and generality to versatile network environments by considering both boards and alarms.
- Formulate the BAPT formation problem as an ILP to obtain the optimal BAPT(s), fulfilling both failure localization and alarm correlation.
- Conduct extensive case studies on single-board and regional failure events to verify the raised BAPT-FL. The results demonstrate its capability of precisely localizing failed device boards in the OTN optical layers.

The remainder of this paper is structured as follows. Section II thoroughly reviews the state-of-the-art alarm correlation approaches in telecommunications. Section III dwells on the proposed BAPT-FL framework. Section IV describes the setup of case studies and results. Section V concludes this paper.

II. Literature Review

Recently, machine learning-based approaches have dominated the realm of alarm correlation [4]. These methods, though, are specific to a network state and may necessitate retraining their models with the changes in network topology or traffic distribution. In [5], [6], the authors quantify the alarm importance with K-means and backpropagation (BP) neural network, where these alarms come from frequent alarm chain sets uncovered by the Apriori algorithm, but they neglect the alarm location information. Liu et al. [7] advance a failure localization framework using the long short-term memory (LSTM) network and BP, aiming at seeking the mapping between those alarming devices and faulty ones. Even so, their method is merely applicable to small-scale static networks that deploy a limited number of lightpaths. Yang et al. [8] address the multi-failure localization problem with the Hopfield neural network. They may have difficulty in establishing relations between the suspected failures and alarms, given that numerous alarms spread across a mass of boards shortly in nowadays OTN. In [9], the researchers transform the single-link failure

localization problem into a binary classification issue regarding root alarms and non-root alarms. A deep belief network is trained to resolve this problem, whereas it is tailored to a particular network state and may not be apt for others. Furthermore, Jia et al. [10] launch their first attempt to use the pre-trained natural language model, called Bidirectional Encoder Representations from Transformers (BERT), to acquire the root alarm and its correlated alarms. However, BERT may struggle against differentiating various alarms with similar semantics. In [11], [12], the single-failure location is inferred from the set of suspected failure locations with a deep neural evolution network, though, it could be time-intensive to create a set of suspected failure locations for each alarm. In [13], the authors adopt two light gradient boosting machine-aided classifiers for pinpointing a single power attenuation failure, but the pre-configured sliding window size may not fit other network environments.

Meanwhile, GNN-based methods have become predominant in alarm correlation, thanks to their additional concern on network topology or correlations among various alarm types. He et al. [14] cluster the status of partial devices with LSTM to vectorize all nodes in the device-level network topology graph, enabling GNN to diagnose the device state and locate the root faulty devices, while it could be intractable to collect a multi-failure dataset for training. In [15]–[17], an alarm knowledge graph (AKG) is built for training a GNN to determine the root alarm(s) and position the single failure, but this AKG primarily includes static knowledge, irrespective of dynamic factors. Further, [18] improves this framework by considering the network topology for handling the multi-failure scenario, which is implemented by combining the AKG and network status KG. Nevertheless, this method overlooks the traffic distribution and has to be retrained to cope with any turbulence in network topology. By deploying optical performance monitors (OPM), the authors in [19], [20] analyze the OPM data with GNN to figure out the most likely failed device from all suspected failed ones, but their model size varies with the strategy of OPM deployment as well as network topology and thus the acquired model may not be generalized to other networks. Afterwards, Zeng et al. [21], [22] employ the graph attention mechanism for processing the power data to localize the single failed component in optical networks. However, the performance of this scheme may be subject to monitoring data loss which is due to any abnormalities in telemetry systems. In [23], a fast data-driven intuitive system using GNN gives several probable faults. A slow knowledge-driven logical system further decides the real fault's location and type from these faults. Still, this system must be retrained when migrating to another network topology. Xu et al. [24] divide the multi-failure localization issue into alarm clustering through linkage prediction and failure localization by inductive representation learning on network topology, but multiple models have to be trained according to different node degrees, therefore being deficient in generality for various topologies. Finally, the authors in [25] gain the root alarms and eliminate the false repair orders with [26], Bayesian network, and GNN, yet their alarm correlation conditions are commonly concluded from experiences and not consolidated as one metric.

III. Proposed BAPT-FL Framework

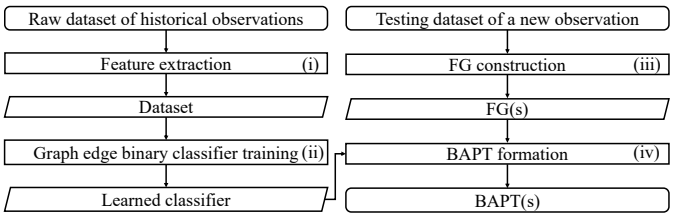


Fig. 2: Flowchart of the proposed BAPT-FL framework.

Fig. 2 shows the flowchart of the proposed BAPT-FL framework. Given the raw dataset of historical observations, we extract the features from each BAPT, as depicted in (i) of Fig. 2 and detailed in Section III-A. Then the obtained dataset is harnessed for training a graph edge binary classifier to weight the edges in each potential BAPT, as exhibited in (ii) of Fig. 2 and elaborated in Section III-B. Further, a new observation is furnished as the testing dataset, aiming at attaining a number of FGs via FG construction, as displayed in (iii) of Fig. 2 and described in Section III-C. The learned classifier and FG(s) are the inputs for the BAPT formation procedure demonstrated in (iv) of Fig. 2 and explained in Section III-D. BAPT-FL culminates in gaining one or multiple BAPTs as the expected output.

A. Feature Extraction

The raw dataset is composed of historical observations collected from a set of OWs, denoted as $\mathcal{O} = \{O_1, \cdots, O_i, \cdots, O_I\}$, $\forall i \in \{1, \cdots, I\}$. During O_i we observe the network topology T_{O_i}, traffic distribution L_{O_i}, alarm set A_{O_i}, failure event, and BAPTs. Here, $T_{O_i} = (B_{O_i}, E_{O_i})$ is an undirected graph indicating the board-level connectivity, where B_{O_i} is the vertex set of device board instances and E_{O_i} is the edge set. L_{O_i} represents a set of lightpaths each traversing several device board instances in B_{O_i}. $A_{O_i} = \{a_1, \cdots, a_{N_{O_i}}\}$ embodies N_{O_i} alarm instances which are sorted in ascending order of their occurrence time. Note that the BAPTs could be inferred by the failure event and B_{O_i} with the assistance of expertise, and we assume that they are consistent with the ground-truth failure-alarm propagation rules. For each BAPT, all its edges are labelled as 1s. Also, we introduce the auxiliary edges for creating the edges labelled as 0s. Some auxiliary edges are carried out by connecting the tree root board instance with each alarm instance, while others are realized by pairing any two alarm instances with a positive occurrence time gap. Eventually, these modified BAPTs constitute the dataset D_{O_i}.

To portray each BAPT in D_{O_i}, we create a feature space $\mathcal{F}$ with dimensions including time, network topology, traffic distribution, and board/alarm attributes, given as below:

- For each vertex within a BAPT, we consider the board type, alarm type, and layer information.

- For each edge within a BAPT, we extract the following features:
  - **occurrence time gap**: the occurrence time gap between two instances;
  - **physical distance**: the length of the shortest path from the instance that happened earlier to the other one in T_{O_i};
  - **hop distance**: the number of OMSs passed by the shortest path in terms of hop count;
  - **lightpath label**: a binary value signifying whether these two instances traverse a common lightpath.

Mapping each BAPT from D_{O_i} to $\mathcal{F}$ results in a transformed dataset $\mathcal{F}(D_{O_i})$. Ultimately, $\forall i \in \{1, \cdots, I\}$, $D = \bigcup_{i=1}^{I} \mathcal{F}(D_{O_i})$ can be garnered as the dataset catering to the need of training the graph edge binary classifier.

B. Graph Edge Binary Classifier

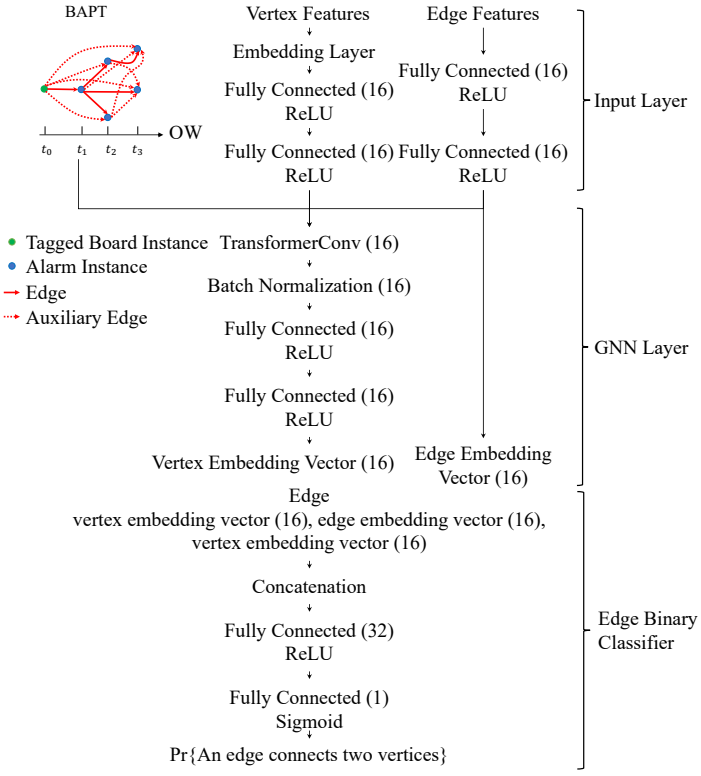


Fig. 3: Architecture of the graph edge binary classifier.

A graph edge binary classifier is designed to evaluate the edge weights for each potential BAPT. We adopt a one-layer GNN whose architecture is shown in Fig. 3, where the dimension of the corresponding layer/embedding vector is annotated in parentheses. The input layer consists of vertex features, edge features, and BAPT. The features of the vertex and edge belong to categorical and numerical ones, respectively. To jointly manipulate these two types of features, the vertex features are mapped onto a continuous space via an embedding layer. Then the vertex and edge features are individually pre-processed with 2 fully connected layers. Each BAPT, which indicates the connectivity amongst these vertices, is treated as an undirected graph due to the fact that the performance of benchmark GNN models on homophilic graphs didn't benefit much from using edge direction [27]. The GNN layer employs 1 convolutional layer with the graph transformer operator [28] that supports message passing with vertex and edge features along the BAPT structure, outputting the vertex embedding vectors. Consequently, a graph edge binary classifier is established to calculate the probability of an edge connecting two vertices as the edge weight. We concatenate the two vertices' embedding vectors as well as the corresponding edge embedding vector and feed it into another fully connected layer.

C. FG Construction

Within each OW P, the testing dataset of a new observation is given, which embraces the network topology T_P, traffic distribution L_P, and alarm set A_P. The presented FG construction process is designed to build a set of FGs, providing the search space for the subsequent BAPT formation. Initially, a device board from T_P is labelled as the tagged board, which is used to generate a subgraph induced by T_P. This subgraph contains all boards going through the tagged board's functional connections, i.e., OTS, OMS, and OCH in the setting of OTN. Then we focus on the alarms reported by the boards in the subgraph and examine whether the tagged board could be associated with the subset of these alarms. Here, BAPTs in the raw dataset can be leveraged for mining the failure-alarm association rules between the tagged board's failure type and alarm type(s) in each BAPT. If the tagged board and its alarm subset satisfy the corresponding association rule, this subgraph is deemed as a valid FG otherwise we skip this tagged board. The above procedure will be performed for each device board in T_P, resulting in a set of FGs, where the tagged board and its alarm subset are attached to each FG as well.

D. BAPT Formation

Given the trained graph edge binary classifier and FGs, the proposed BAPT formation algorithm aims to construct the minimum cost BAPTs while covering all alarm instances in A_P, with each BAPT rooted by a failed board and taking some alarms as the leaves.

Fig. 4 showcases the proposed BAPT formation algorithm. For each FG, a potential BAPT can be generated by pairing its tagged board with each alarm from the alarm subset and connecting all alarm instance pairs with a positive occurrence time gap. Then each potential BAPT is fed into the trained graph edge binary classifier for weighting its edges. These weighted potential BAPTs will be post-processed with two steps. We take the graph union of all weighted potential BAPTs, followed by checking whether the resultant graph is a simple graph. If there exists more than one edge between any two vertices, we merely reserve one of the maximum-weight edges. Afterwards, a weighted directed acyclic graph $\mathcal{G}_P$ is prepared as the search space of optimal BAPT(s), denoted as $\mathcal{G}_P = (V_P, E_P, C(E_P))$. $V_P = B_P \bigcup A_P$ represents

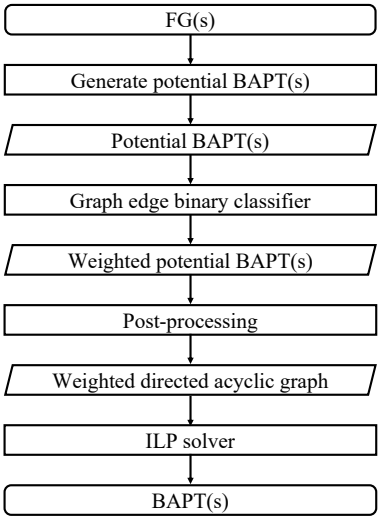


Fig. 4: Flowchart of the proposed BAPT formation algorithm.

the vertex set aggregated by the board instance set B_P and alarm set A_P. $E_P = E_P^1 \bigcup E_P^2$ is the edge set, where $E_P^1 = B_P \times A_P$, $E_P^2 \subseteq A_P \times A_P$ are the edge sets formed by boards and alarms as well as alarm pairs, respectively. Also, $C(E_P)$ is the set of non-negative edge costs, where $\forall v_i, v_j \in V_P, (v_i, v_j) \in E_P, c_{ij} \in C(E_P)$ signifies the weight/cost of edge (v_i, v_j) and it's defined in (1):

$$c_{ij} = \begin{cases} 1 - Pr\{v_i \text{ fails}\} \\ \quad \cdot Pr\{(v_i, v_j) \text{ connects } v_i, v_j\}, \text{if } (v_i, v_j) \in E_P^1, \\ 1 - Pr\{(v_i, v_j) \text{ connects } v_i, v_j\}, \text{if } (v_i, v_j) \in E_P^2, \end{cases} \tag{1}$$

where $Pr\{v_i \text{ fails}\}$ is the probability that board instance v_i fails. It can be derived from the probability density function (PDF) of the corresponding board's time-to-failure, where the PDF could be estimated grounded in historical failure events taking place at this board. $Pr\{(v_i, v_j) \text{ connects } v_i, v_j\}$ is the probability output from the trained graph edge binary classifier.

Furthermore, the problem of coming up with the optimal BAPTs $\mathcal{G}_P^*$ can be framed as an ILP problem defined in (2):

$$\text{minimize} \quad \sum_{e \in E_P} c_e x_e + \sum_{u \in B_P} (1 - Pr\{u \text{ fails}\}) \cdot y_u \tag{2a}$$

$$\text{subject to} \quad \sum_{e \in \delta^-(v)} x_e = 1, \quad \forall v \in A_P, \tag{2b}$$

$$x_e \leq y_u, \quad \forall u \in B_P, \forall e \in \delta^+(u), \tag{2c}$$

$$x_e \in \{0, 1\}, \quad \forall e \in E_P, \tag{2d}$$

$$y_u \in \{0, 1\}, \quad \forall u \in B_P. \tag{2e}$$

$\forall e \in E_P, \forall u \in B_P$, two binary variables x_e, y_u are defined, where x_e takes 1 if the edge e is selected by $\mathcal{G}_P^*$ and 0 otherwise; while y_u takes 1 if the board instance vertex u is chosen as a tree root in $\mathcal{G}_P^*$ and 0 otherwise. The objective function (2a) aims to extract $\mathcal{G}_P^*$ that minimizes the total cost of opted edges and board instances. Constraint (2b) implies that for each alarm instance vertex v, only one item from the set of edges incoming to v, namely $\delta^-(v)$, must be chosen to ensure all alarm instances are visited by $\mathcal{G}_P^*$. Constraint (2c) suggests that for each board instance vertex u, if any item from the set of edges outgoing from u, called $\delta^+(u)$, is chosen, then u must be selected.

By solving the above ILP, the anticipated BAPT(s) $\mathcal{G}_P^*$ is gained, which achieves both failure localization and alarm correlation.

IV. Case Studies

Extensive case studies are performed to validate the viability of the proposed BAPT-FL in the OTN optical layers and compare it with the counterparts. Firstly, we develop an OTN simulator [29] [30] to harvest ground-truth alarms and BAPTs according to the given failure event, failure-alarm propagation rule database, as well as the network topology and traffic distribution during an OW. The rule database incorporates 38 entries in the form of *One2Many*, where 16 failure types, 16 board types, and 25 alarm types are taken into account. Without loss of generality, each failure event independently hits a certain number of boards and breaks their traversing functional connections. Besides, we assume that the failure-alarm propagation process always succeeds even though any failed board exists along the notification signal propagation path. On the other hand, we suppose that each board instance is equiprobable to fail, thereby ignoring the cost of choosing board instances by setting $Pr\{u \text{ fails}\}$ as 1 in (2a).

The case studies aim to verify the generality and migratability of BAPT-FL. For this purpose, we apply the graph edge binary classifier trained by the raw dataset to the testing datasets gathered from OTN environments with diverse failures, network topologies, traffic distributions, and noise alarms. The state-of-the-art counterparts for comparison include BP [7], LSTM [7], convolutional neural network (CNN) [9], BERT [10], GNN [15]–[17], GNN-Dual [23], and IC-FD [31].

A. Setup

1) Raw Dataset: The raw dataset is produced from a default network state. It comprises 8 nodes, 264 boards, and 20 lightpaths each going through 14 boards on average and its board-level average degree is 2.2. We collect 3086 alarms from this default network state via 264 1-minute OWs, where these OWs are initiated by the single-board failure events that in turn hit each board in the network topology.

2) Dataset: The raw dataset leads to the dataset of 264 BAPTs for training the graph edge binary classifier. The dataset is split into training and validation sets with a ratio of 9:1. We utilize the min-max scaling to normalize the numerical features. The binary cross entropy loss function is optimized with Adam at a learning rate of 0.005. The batch size and number of epochs are set to 64 and 500, respectively. In addition, we monitor the value of the area under the curve on the validation set in each epoch and make use of early stopping [32] for mitigating overfitting.

3) Machine Learning Model Counterparts: The network architectures of machine learning model counterparts under the raw dataset are concisely delivered as follows. For BP and LSTM, two networks of dimension $120 \times 64 \times 32 \times 264$ and $120 \times 64 \times 264$ are established. The CNN model is made up of 25 input layer units and 3 hidden layers whose number of neurons are 256, 128, and 32, where the kernel size in each hidden layer is 3×3. As for BERT, it encodes the alarm contextual information with a 768-dimension vector and sets the length of each alarm transaction to 3. For training GNN, an AKG with 41 entities is structured based on the rule database. For GNN-Dual, one convolutional layer using the graph operator in [33] is utilized to obtain a 32-dimension feature vector for each node. Lastly, IC-FD exerts a 3-layer fully connected neural network whose number of neurons are 32, 32, and 64, respectively.

4) Testing Datasets: The process of generating testing datasets is sketched out as follows. To assess the benchmark performance for each scheme, we emulate a single-board failure dataset of 10 failed boards randomly selected from the default network state.

Moreover, we generate two regional failure datasets to evaluate the generality and migratability of each method. One dataset is composed of regional failures from 10 new network states achieved by altering the number of lightpaths in a common network topology. This network topology consists of 10 nodes and 2380 boards with a board-level average degree of 2.06. The number of lightpaths ranges from 10 to 100 with an equal interval of 10 and each lightpath averagely traverses 14 boards. The other dataset contains the regional failures from 10 new network states each taking 40 lightpaths. These network states correspond to different topological sizes, where the number of nodes, number of boards, board-level average degree, and average number of boards traversed by each lightpath vary from 11 to 20, 461 to 700, 2.21 to 2.26, and 12 to 18, respectively. Note that each regional failure starts with selecting one board from the network topology at random, which is further expanded by arbitrarily adding one board at a time and ensuring that the chosen boards are connected. The number of failed boards in each regional failure changes from 1 to 10 and the average degree of each regional failure is 1.39.

In the end, the single-board failure dataset incorporating 10 failed boards is simulated to test BAPT-FL's noise-resistant ability. For each failed board, different ratios of noise alarms are introduced on top of the true alarms.

5) ILP Solver and Performance Metrics: To deal with the ILP problem with a huge number of variables, we take the branch-and-price method [34], which is implemented in the high-performance optimization software (HiGHS) [35] that is dedicated to large-scale sparse linear optimization models.

The performance metrics called *F-measure(F)* and *F-measure(R)* are considered for evaluating the two outcomes from our framework in each imitated instance, termed *failed boards* and *root alarms*. *F-measure(F)* and *F-measure(R)* are defined in (3) and (4), where a larger value implies better performance in identifying failed boards/root alarms:

$$F\text{-}measure(F) = 2 \cdot \frac{Precision(F) \cdot Recall(F)}{Precision(F) + Recall(F)}, \tag{3}$$

$$Precision(F) = \frac{\#\text{ of correctly identified failed boards}}{\text{total }\#\text{ of identified failed boards}},$$

$$Recall(F) = \frac{\#\text{ of correctly identified failed boards}}{\text{total }\#\text{ of failed boards}},$$

$$F\text{-}measure(R) = 2 \cdot \frac{Precision(R) \cdot Recall(R)}{Precision(R) + Recall(R)}, \tag{4}$$

$$Precision(R) = \frac{\#\text{ of correctly identified root alarms}}{\text{total }\#\text{ of inferred root alarms}},$$

$$Recall(R) = \frac{\#\text{ of correctly identified root alarms}}{\text{total }\#\text{ of root alarms}}.$$

B. Results

1) Graph Edge Binary Classifier: The training curve of the graph edge binary classifier is shown in Fig. 5. The loss converges to 0.0862/0.0891 and the accuracy reaches 96.07%/95.75% after 463 epochs on the training/validation set.

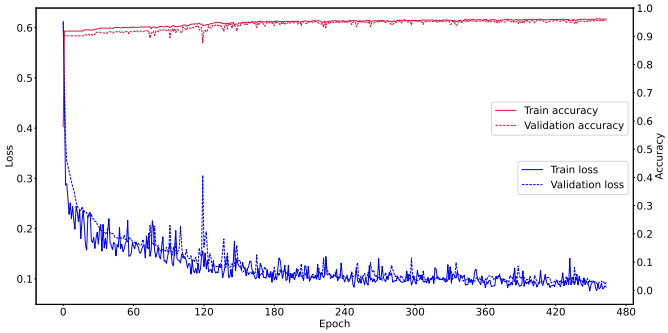


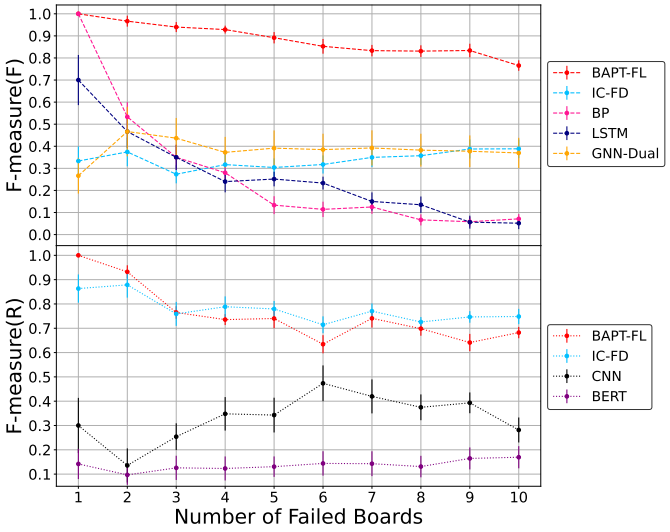
Fig. 5: Training curve of the graph edge binary classifier.

TABLE I: Results on the single-board failure dataset for benchmark performance comparison

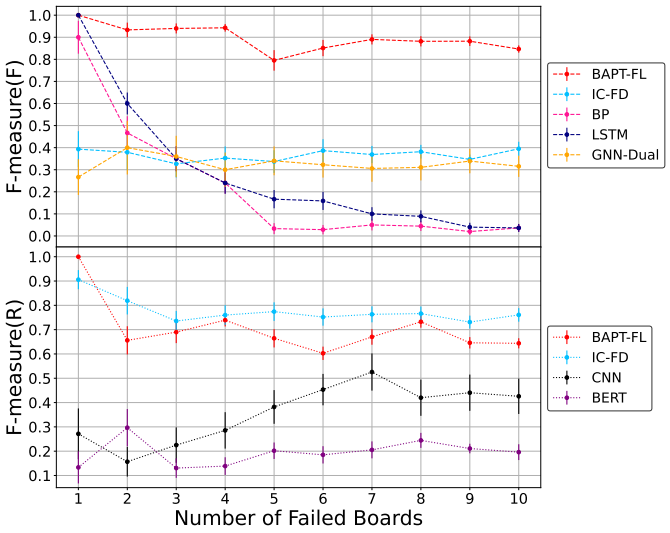
Method	# of trainable parameters	*F-measure(F)*	*F-measure(R)*
BAPT-FL	4521	1.000±0.000	1.000±0.000
IC-FD	5585	0.391±0.211	0.747±0.241
CNN	338378	N/A	0.630±0.272
BERT	2521646		0.200±0.211
BP	40904	0.500±0.310	N/A
LSTM	64520	0.500±0.310	
GNN	51999	0.800±0.248	
GNN-Dual	16801	0.2567±0.17	

2) BAPT Formation: Firstly, the results of all schemes on the single-board failure dataset for benchmark performance comparison are listed in Table I, where the value before "±" means the average *F-measure(F)/F-measure(R)* over 10 failure events and the value after "±" stands for the margin of error at the confidence interval of 95%. Notably, BAPT-FL attains a pronounced advantage over its counterparts in all employed metrics while taking the fewest trainable parameters.

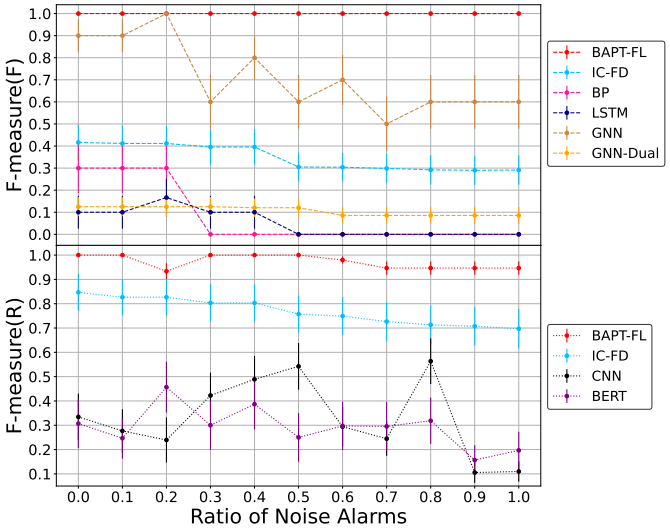
Furthermore, the results of all schemes on two regional failure datasets are summarized in Fig. 6a and Fig. 6b. Obviously, BAPT-FL performs the best among all its counterparts in



(a) Result under various traffic distributions.



(b) Result under various network topologies.



(c) Result under various ratios of noise alarms.

Fig. 6: Results on two regional failure datasets and a single-board failure dataset with noise alarms.

recognizing failed boards. Nevertheless, BP and LSTM grapple with a significant performance decline with increasing the number of failed boards, which hints that they could lose more real failed boards on account of neglecting abundant alarm attributes. Again, BAPT-FL outperforms its counterparts in terms of root alarm identification except for IC-FD. However, CNN and BERT suffer from dramatic performance degradation because they completely ignore the traffic distribution and spatial relation among these alarms. Similarly, GNN-Dual is subject to poor performance due to overlooking the traffic distribution, even if the network topology is considered. Note that although IC-FD is on par with or even better than BAPT-FL in recognizing root alarms, it's still worse than BAPT-FL in localizing failed boards since it independently evaluates each edge weight without considering the hidden tree-shape correlation pattern between the tagged board and alarm subset.

Additionally, the results of various schemes on the single-board failure dataset with noise alarms are exhibited in Fig. 6c. BAPT-FL has the strong capability of noise immunity thanks to its careful consideration of the correlation among the received alarms. In contrast, all its counterparts are severely degraded in detecting failed boards/root alarms with the increase in the ratio of noise alarms.

Further note that since the location information of failed boards/alarms changes through the evolving network environments, all models used by BAPT-FL's counterparts except IC-FD require retraining when any disturbance occurs in the network topology or traffic distribution, but BAPT-FL needs to be trained once with the raw dataset gathered from the default network state and thus it realizes the best migratability among all considered counterparts.

3) Complexity Analysis: We assume that during an OW P, the board set $B_P = \{b_1, \cdots, b_{M_P}\}$ and alarm set $A_P = \{a_1, \cdots, a_{N_P}\}$ are provided. The step of FG construction requires examining each board from B_P, leading to m_P FGs and the computation complexity of $\mathcal{O}(M_P)$. Furthermore, the complexity of the BAPT formation algorithm derives from using the graph edge binary classifier and ILP solver. The former one results in $\mathcal{O}(m_P)$. The latter one is known as an NP-hard problem and thus it's overlooked in the complexity analysis, where the number of variables and constraints considered by the ILP in (2) encompass no more than $m_P + \frac{N_P \cdot (N_P - 1)}{2}$ and $N_P + m_P \cdot N_P$, respectively.

V. Conclusion

This paper presents a novel framework for board-level failure localization in the OTN, called Board-Alarm Propagation Tree based Failure Localization (BAPT-FL). The proposed BAPT-FL is characterized by several impressive modelling approaches. Firstly, we have collaborated boards and alarms into the correlation process that pinpoints both failed boards and root alarms. Secondly, we have trained a graph edge binary classifier that is generalized to various network environments that deploy the same failure-alarm propagation rules. Thirdly, we have proposed the concept of BAPT that effectively models the correlation among the failed boards and alarms, and the

BAPT formation procedure is formulated as a solvable ILP problem that provides a graceful solution.

Extensive case studies are conducted to compare the proposed BAPT-FL with other schemes in diverse network environmental changes. We observe that BAPT-FL has achieved extraordinary and stable performance in localizing failed boards and root alarms, which champions our claim that BAPT-FL boasts sufficient generality and migratability in variable network environments.

REFERENCES

[1] *Recommendation G.709/Y.1331 (06/20)*. ITU-T Rec. [Online]. Available: https://www.itu.int/rec/T-REC-G.709-202006-I/en

[2] *Optical Transport Network (OTN) tutorial*. ITU-T. [Online]. Available: https://www.itu.int/ITU-T/studygroups/com15/otn/OTNtutorial.pdf

[3] *OSN 8800 6800 3800 V100R009C10 Hardware Description 03*. Huawei. [Online]. Available: https://support.huawei.com/enterprise/en/doc/EDOC1000079083

[4] D. Wang, C. Zhang, W. Chen, H. Yang, M. Zhang, and A. P. T. Lau, "A review of machine learning-based failure management in optical networks," *Science China Information Sciences*, vol. 65, no. 11, pp. 1–19, 2022.

[5] D. Wang, L. Lou, M. Zhang, A. C. Boucouvalas, C. Zhang, and X. Huang, "Dealing with alarms in optical networks using an intelligent system," *IEEE Access*, vol. 7, pp. 97760–97770, 2019.

[6] M. Zhang and D. Wang, "Machine learning based alarm analysis and failure forecast in optical networks," in *2019 24th OptoElectronics and Communications Conference (OECC) and 2019 International Conference on Photonics in Switching and Computing (PSC)*, Fukuoka, Japan, pp. 1–3, 2019.

[7] T. Liu, H. Mei, Q. Sun, and H. Zhou, "Application of neural network in fault location of optical transport network," *China Communications*, vol. 16, no. 10, pp. 214–225, 2019.

[8] H. Yang, B. Wang, Q. Yao, A. Yu, and J. Zhang, "Efficient hybrid multi-faults location based on hopfield neural network in 5G coexisting radio and optical wireless networks," *IEEE Transactions on Cognitive Communications and Networking*, vol. 5, no. 4, pp. 1218–1228, 2019.

[9] A. Yu, H. Yang, Q. Yao, Y. Li, H. Guo, T. Peng, H. Li, and J. Zhang, "Accurate fault location using deep belief network for optical fronthaul networks in 5G and beyond", *IEEE Access*, vol. 7, pp. 77932–77943, 2019.

[10] J. Jia, D. Wang, C. Zhang, H. Yang, L. Guan, X. Chen, and M. Zhang, "Transformer-based Alarm Context-Vectorization Representation for Reliable Alarm Root Cause Identification in Optical Networks," in *2021 European Conference on Optical Communication (ECOC)*, Bordeaux, France, pp. 1–4, 2021.

[11] X. Zhao, H. Yang, H. Guo, T. Peng, and J. Zhang, "Accurate fault location based on deep neural evolution network in optical networks for 5G and beyond," in *2019 Optical Fiber Communications Conference and Exhibition (OFC)*, San Diego, CA, USA, pp. 1–3, 2019.

[12] H. Yang, X. Zhao, Q. Yao, A. Yu, J. Zhang, and Y. Ji, "Accurate fault location using deep neural evolution network in cloud data center interconnection," *IEEE Transactions on Cloud Computing*, vol. 10, no. 2, pp. 1402–1412, 2022.

[13] J. Babbar, A. Triki, R. Ayassi, and M. Laye, "Machine learning models for alarm classification and failure localization in optical transport networks," *Journal of Optical Communications and Networking*, vol. 14, no. 8, pp. 621–628, 2022.

[14] J. He and H. Zhao, "Fault diagnosis and location based on graph neural network in telecom networks," in *2020 International Conference on Networking and Network Applications (NaNA)*, Haikou, China, pp. 304–309, 2020.

[15] Z. Li, Y. Zhao, Y. Li, S. Rahman, Y. Wang, X. Yu, L. Zhang, G. Feng, and J. Zhang, "Demonstration of alarm knowledge graph construction for fault localization on ONOS-based SDON platform," in *2020 Optical Fiber Communications Conference and Exhibition (OFC)*, San Diego, CA, USA, pp. 1–3, 2020.

[16] Z. Li, Y. Zhao, Y. Li, S. Rahman, X. Yu, and J. Zhang, "Demonstration of fault localization in optical networks based on knowledge graph and graph neural network," in *2020 Optical Fiber Communications Conference and Exhibition (OFC)*, San Diego, CA, USA, pp. 1–3, 2020.

[17] Z. Li, Y. Zhao, Y. Li, S. Rahman, F. Wang, X. Xin, and J. Zhang, "Fault localization based on knowledge graph in software-defined optical networks," *Journal of Lightwave Technology*, vol. 39, no. 13, pp. 4236–4246, 2021.

[18] J. Huang, Z. Chen, B. Zhu, Y. Jing, Y. Liang, X. Yang, R. Li, J. Han, and Y. Zhao, "Multi-fault localization based on knowledge graph combined with network topology in optical networks," in *Ninth Symposium on Novel Photoelectronic Detection Technology and Applications*, vol. 12617, pp. 1093–1098, 2023.

[19] R. Wang, J. Zhang, S. Yan, C. Zeng, H. Yu, Z. Gu, B. Zhang, T. Taleb, and Y. Ji, "Suspect fault screening assisted graph aggregation network for intra-/inter-node failure localization in ROADM-based optical networks," in *2022 European Conference and Exhibition on Optical Communication (ECOC)*, Basel, Switzerland, pp. 1–4, 2022.

[20] R. Wang, J. Zhang, S. Yan, C. Zeng, H. Yu, Z. Gu, B. Zhang, T. Taleb, and Y. Ji, "Suspect fault screen assisted graph aggregation network for intra-/inter-node failure localization in ROADM-based optical networks," *Journal of Optical Communications and Networking*, vol. 15, no. 7, pp. C88–C99, 2023.

[21] C. Zeng, J. Zhang, R. Wang, B. Zhang, and Y. Ji, "Component fault location in optical networks based on attention mechanism with monitoring data," in *2022 European Conference on Optical Communication (ECOC)*, Basel, Switzerland, pp. 1–4, 2022.

[22] C. Zeng, J. Zhang, R. Wang, B. Zhang, and Y. Ji, "Multiple attention mechanisms-driven component fault location in optical networks with network-wide monitoring data," *Journal of Optical Communications and Networking*, vol. 15, no. 7, C9–C19, 2023.

[23] J. Ji, F. Zhu, J. Cui, H. Zhao, and B. Yang, "A dual-system method for intelligent fault localization in communication networks," in *ICC 2022-IEEE International Conference on Communications*, Seoul, Korea, Republic of, pp. 4062–4067, 2022.

[24] X. Xu, H. Chen, J. E. Simsarian, R. Ryf, M. Mazur, L. Dallachiesa, N. K. Fontaine, and D. T. Neilson, "Optical Network Diagnostics Using Graph Neural Networks and Natural Language Processing," in *2023 Optical Fiber Communications Conference and Exhibition (OFC)*, San Diego, CA, USA, pp. 1–3, 2023.

[25] W. Jiang and Y. Bai, "APGNN: Alarm Propagation Graph Neural Network for fault detection and alarm root cause analysis," *Computer Networks*, vol. 220, pp. 109485, 2023.

[26] X. Zhang, Y. Bai, P. Feng, W. Wang, S. Liu, W. Jiang, J. Zeng, and R. Wang, "Network alarm flood pattern mining algorithm based on multi-dimensional association," in *Proceedings of the 21st ACM International Conference on Modeling, Analysis and Simulation of Wireless and Mobile Systems*, New York, NY, USA, pp. 71–78, 2018.

[27] E. Rossi, B. Charpentier, F. Di Giovanni, F. Frasca, S. Günnemann, and M. Bronstein, "Edge Directionality Improves Learning on Heterophilic Graphs," 2023, arXiv:2305.10498.

[28] Y. Shi, Z. Huang, S. Feng, H. Zhong, W. Wang, and Y. Sun, "Masked label prediction: Unified message passing model for semi-supervised classification," 2020, arXiv:2009.03509.

[29] Z. Li, "Design of an OTN-based Failure/Alarm Propagation Simulator," M.S. Thesis, Electr. and Comput. Eng., Univ. of Waterloo, Canada, 2022. [Online]. Available: http://hdl.handle.net/10012/18304.

[30] Z. Li, P. -H. Ho, Y. Jiao, B. Li, and Y. You, "Design of an OTN-based Failure/Alarm Propagation Simulator," in *2022 International Conference on Networking and Network Applications (NaNA)*, Urumqi, China, pp. 1–5, 2022.

[31] Y. Jiao, P. -H. Ho, X. Lu, K. Liang, Y. You, J. Tapolcai, B. Li, and L. Peng, "On Real-time Failure Localization via Instance Correlation in Optical Transport Networks," in *2023 IFIP Networking Conference (IFIP Networking)*, Barcelona, Spain, pp. 1–9, 2023.

[32] Y. Bengio, "Practical recommendations for gradient-based training of deep architectures," *Neural Networks: Tricks of the Trade: Second Edition*, Berlin, Heidelberg: Springer, 2012.

[33] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," 2016, arXiv:1609.02907.

[34] C. Barnhart, E. L. Johnson, G. L. Nemhauser, M. W. Savelsbergh, and P. H. Vance, "Branch-and-price: Column generation for solving huge integer programs," *Operations research*, vol. 46, no. 3, pp. 316–329, 1998.

[35] Q. Huangfu and J. J. Hall, "Parallelizing the dual revised simplex method," *Mathematical Programming Computation*, vol. 10, no. 1, pp. 119–142, 2018.