On Real-time Failure Localization via Instance Correlation in Optical Transport Networks

Yan Jiao*, Pin-Han Ho*, Xiangzhu Lu*, Kairan Liang*, Yuren You†,
János Tapolcai‡, Bingbing Li§, and Limei Peng¶
*Department of Electrical and Computer Engineering, University of Waterloo, Waterloo, Canada
†Huawei Ottawa Research and Development Centre, Ottawa, Canada
‡Department of Telecommunications and Media Informatics, Budapest University of Technology and Economics, Hungary
§School of Computer Science, Hubei University of Technology, Wuhan, China
¶School of Computer Science and Engineering, Kyungpook National University, Daegu, South Korea
Email: *{y42jiao, p4ho, x244lu, k4liang}@uwaterloo.ca, †yuren.you@huawei.com,
‡tapolcai@tmit.bme.hu, §bbli@hbut.edu.cn, ¶auroraplm@knu.ac.kr

***Abstract*—Failure localization serves as a key to an effective fault management plane in the Internet backbone. This paper investigates a novel failure localization approach, namely Instance Correlation based Fault Diagnosis (IC-FD), for achieving efficient fault management in Optical Transport Networks (OTN). The IC-FD is aimed at real-time localization of failed components in the optical layer of OTN through correlation of alarms and status changes of network devices (referred to as *instances*) via a learned binary classifier. The outcome of IC-FD is one or multiple instance correlation trees (ICT) where the instances corresponding to the faulty network devices are taken as the tree roots. Notably, the proposed binary classifier is characterized by an intelligent feature extraction of historical instance correlation in dimensions of time, board/alarm attribute, network topology, and traffic distribution. Extensive case studies are conducted to demonstrate the advantages gained by IC-FD in terms of its high precision and low computation complexity, as well as analysis of its performance due to various environmental turbulence such as network topology, traffic diversity and noise alarms.**

***Index Terms*—failure localization, correlation analysis, similarity learning, optical transport networks (OTN)**

I. Introduction

Telecommunication networks, particularly the Internet optical backbones, have gone through significant expansions in the past decades not only in their capacity and geographical coverage, but also in their heterogeneous nature where a multi-service and multi-tenant environment is supported. Optical transport network (OTN) is a standard control and management framework under ITU-T that serves as the basis of facilitating such expansions by enabling various service flows multiplexed via individual optical flows [1]. The OTN control plane provides a suite of rigid alarming mechanisms at each device and fiber segment (generally termed *board* in the following context) in response to any failure event detected by the sensor associated with the board. For example, a transponder board triggers an alarm reported to the network management system (NMS) when any irregularity affecting the quality of the received lightpath is identified. Another example is that the failure of a fiber segment board would propagate to the receiving fiber interface unit (FIU) board that in turn reports an alarm to the NMS.

In general, a failure event could happen at any board that unexpectedly affects the optical signals traversing through the board. At this moment, the failure event may *propagate* through multiple boards in vicinity and/or that in geographically remote areas due to the traffic distribution. In addition, an alarm may be triggered at a board not only due to an identified failure event, but also in response to a notification alarm issued by another remote board. The two sources of alarms could cause a vast number of alarms that significantly boost the complexity of alarm correlation and failure localization. This situation becomes even worse because of the large geographical coverage and huge number of network entities of the current Internet backbone.

Alarm correlation has been considered an effective approach to achieve the required precision in identifying dependency for each pair of collected alarms. With those dependencies, most dependent alarms can be removed such that the failure event(s) can be inferred/pinpointed with much reduced complexity.

An example is given in Fig. 1 where five boards A, B, C, D, and E are connected by corresponding fiber pairs. As shown in Fig. 1(a), let the fiber cut event on the link from D to B be denoted as f_1, which is firstly detected by B, noted as an event f_2, and triggering an alarm a_2 reported to the NMS. Upon f_2, B notifies its neighbouring boards D and E as events f_3 and f_4, respectively, where the incurred alarms a_3 and a_4 are reported to the NMS accordingly. The entire alarm propagation process is represented as an instance correlation tree (ICT) shown in Fig. 1(b). Note that although f_1 does not correspond to any alarm reported to the NMS, it is the root cause taken a further inference to localize. Another example is given in Fig. 1(c) where failure on board D, noted as f_5, has even disabled its sensor and thus reported no alarm to the NMS. The failure event f_5 propagated to boards A, B, and C due to the commonly traversing traffic (i.e., an OTS, OMS, or OCH connection) that caused events f_6, f_7, and f_8, which further triggered alarms a_6, a_7, and a_8 reported to the

ISBN 978-3-903176-57-7 ©2023 IFIP

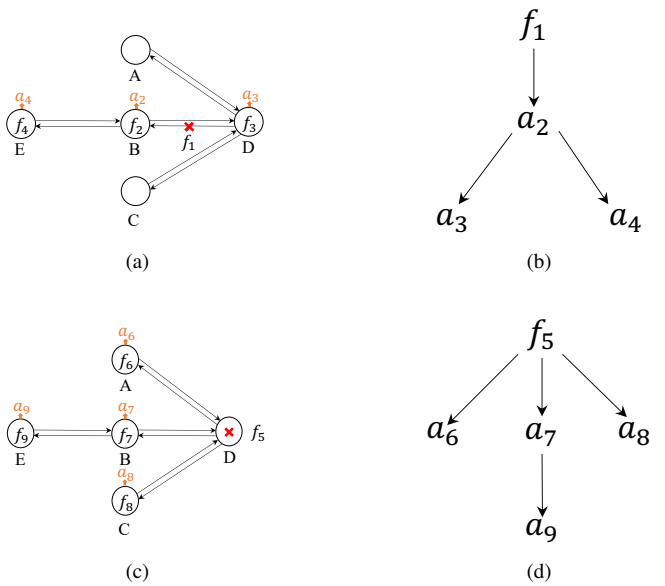


Fig. 1. (a), (c) Examples of different failure events hitting the same network, where the red cross indicates the faulty fiber segment/board. (b), (d) Corresponding ICTs.

NMS, respectively. At the same time, board E had event f_9 to occur due to the notification by B and reported an alarm a_9. The expected result of failure localization is by correlating the alarms a_6, a_7, a_8, and a_9, by which the NMS has to come up with the ICT as shown in Fig. 1(d) in spite of the fact that the root cause f_5 is completely "silent" throughout the whole alarm propagation and reporting process.

In addition to precision, the desired features of failure localization design include sufficient generality to various network environments and adaptability to the changing network status, including any possible variation in network topology and traffic distribution. Further, scalability and computational efficiency should be pursued such that the NMS can swiftly identify the observed irregularity and launch the required reaction/restoration to the incurred damages.

Motivated by its importance and stringent requirements, this paper introduces a novel failure localization algorithm in the optical layer of OTN, called Instance Correlation based Fault Diagnosis (IC-FD), aiming to explore various design dimensions for achieving all the desired features. We firstly define *board instances* and *alarm instances*, which represents the status of each board and the collected alarms at the NMS during an observation window, respectively. By assuming a functioning board can become faulty at most once at the beginning of an observation window, a board instance can be at most directly correlated with a set of alarm instances in time vicinity; while an alarm instance contains a number of features related to the reported alarm.

By taking each instance as a vertex, the correlation of an instance pair is nothing but the likelihood of the existence of an arc interconnecting the two vertices. As such we investigate the instance similarity measurement using a machine learning approach by jointly considering the network topology and dynamic traffic distribution, where the trained binary classifier is migratable to any possible network environment with the same alarm generation/propagation rules. With all the labeled arcs, *ICT formation* can be exclusively completed by heuristically solving an integer linear programming (ILP) problem, where each ICT has a board instance as the tree root connecting to one or multiple alarm instances. The goal of the ICT formation is to cover all the alarms by the ICTs where each ICT demonstrates a complete alarm propagation process due to the faulty board.

The contributions of this paper are summarized as follows.

- Investigate a novel failure localization approach, namely IC-FD, which relies on a machine learning-based binary classifier and ICT formation modeling approach.
- Propose a novel binary classifier model aiming to achieve the best generality and adaptation to the versatile network environment by taking both board instances and alarm instances as the input of the model.
- Introduce a novel ICT formation process for obtaining the best possible ICTs according to the given set of alarms.
- Conduct extensive case studies to verify the proposed IC-FD approach and show that it can achieve real-time and precise failure localization in the optical layer of OTN.

The rest of this paper is organized as follows. Section II is on the literature review. Section III presents the system model, followed by the proposed IC-FD approach in Section IV. Section V presents the case study setup and the results. Section VI concludes the paper.

II. Literature Review

Failure localization based on root-cause alarm analysis can be generally achieved via alarm correlation. An expert system called IMPACT was firstly raised in [2] [3] for a number of functions supporting real-time network management, namely intelligent alarm filtering, alarm generalization, and fault diagnosis. The expert system approach nonetheless relies on a knowledge base created by domain experts and could be subject to intolerably high complexity and maintenance costs. A dependency graph approach was considered in the studies [4] [5], where a dependency graph of various types of events, mostly Bayesian networks, was constructed according to the log files without considering network topological or traffic information. Thus, the obtained result according to a specific network at a given moment may not be migratable to another. Extensive research efforts have taken data-driven approaches. Some of them employed pattern mining techniques, mostly inspired by the pioneering research work called TASA [6] [7], including the association rule mining [8] [9], frequent episode mining [10] [11], and sequential patterning mining [12] [13]. Some others such as [14] [15] have employed K-means and artificial neural network (ANN) to quantify the alarm importance, where a rule mining algorithm weighted by the alarm importance was applied to discover alarm correlation rules. Note that all the above mentioned methods have focused

on temporal relation and type information of the collected alarms while completely ignoring their spatial relations.

Some techniques based on artificial intelligence (AI) have been leveraged for alarm root cause analysis [18]. In [19], the long-short term memory network (LSTM) was used to locate the fault, whose output adopted fuzzy theory to represent the possible fault location with the probability from 0 to 1. But this model can only be applied to a small-scale static network with a very limited number of lightpaths, and it needs to be retrained in response to the changes in network topology and/or traffic distribution with the new fault alarm dataset. In [20] [21], a deep neural evolution network (DNEN) was introduced to handle large-scale alarm sets by creating a mapping to a fault set via global search. The obtained model is specific to the given network topology and does not consider traffic distribution. In [22]–[24], an alarm knowledge graph (KG) was built for alarm relation reasoning. Then a graph neural network (GNN) was trained with this KG for inferring alarm relations and root cause alarm(s). Here, the KG only incorporates static knowledge regarding correlations among faults and alarms without accommodating any dynamic scenario of alarm correlation, where alarms are propagated along certain static connections. In [25], the alarm context was vectorized with the pre-trained bidirectional encoder representations from transformers (BERT), and the Transformer Encoder was used to identify root cause alarm(s). It could turn out to be a challenge for BERT in distinguishing various alarm types whose semantics are very similar. Authors in [22]–[25] assumed that the fault locations were deduced according to the location of root alarm(s), which isn't necessarily true based on the aforementioned examples in Fig. 1.

In [16], the occurrence of a type A alarm is likely to trigger a type B alarm with the confidence evaluated by an asymmetric measure in a dynamic attributed graph that incorporates the network topology and collected alarms. In [17], the authors consider the fact that the alarm sequences generated by nodes that are topological neighbours are no longer independent. Although effective in perspective scenarios, both [16] and [17] are subject to a number of issues. Firstly, their models don't consider the facts that alarm propagation mostly occurs along certain connections (e.g., OCH in OTN), and that the information of instantaneous network traffic distribution may solidly facilitate the desired alarm correlation process. Secondly, to improve the computational efficiency, [16] assumes that alarms are only correlated within a fixed-size time window whereas [17] prefers a smaller value for the farthest topological distance between a pair of correlated alarms. However, it lacks a general rule to define the parameters for scaling the window size and the maximum hops. Lastly, both schemes need a significant amount of data to obtain sufficient statistics for alarm correlation, and may not be applicable to the network environment such as OTN optical layer.

III. Failure/Alarm Propagation Model

In this study, two types of instances are defined. An *alarm instance* is denoted by a 5-tuple $a_i = \{t_i, h_i, b_i, m_i, r_i\}$, $\forall i \in \{1, \cdots, N\}$, where t_i is the occurrence time, h_i and b_i are the ID and type of the board that issues this instance, respectively. m_i is the alarm type and r_i is the layer information that can be either OTS, OMS, or OCH. A *board instance* is denoted by a 2-tuple $f_j = \{h_j, b_j\}$, $\forall j \in \{1, \cdots, M\}$, where h_j is the board ID and b_j is the board type.

The correlation of two instances is evaluated between every pair of instances. An instance correlation between f_j and a_i could be due to the fact that the failure on board instance f_j triggers an alarm instance a_i, which is denoted as $f_j \rightarrow a_i$. Another possible scenario of instance correlation is between a_i and $a_{i'}$, where the alarm instance corresponding to a_i triggers another alarm instance $a_{i'}$, which is denoted as $a_i \rightarrow a_{i'}$.

In the setting of OTN, a board that has been hit by a failure/received a notification signal would change the status of board instance f_j/initiate alarm instance a_i, and this board could notify a remote board of reporting another alarm instance $a_{i'}$ to the NMS. The direction of failure/alarm propagation can be divided into *forward propagation* (FP), *backward propagation* (BP), and *local notification* (LN). With FP (or BP), the source board initiates an instance f_j/a_i and sends a notification signal to the destination board that in turn emits another $a_{i'}$. Specifically, the FP follows the same direction of the corresponding optical flow, while the BP occurs in the reverse direction. The LN, on the other hand, must be a single alarm instance reported by a board without any alarm propagation process.

Accordingly, the failure/alarm propagation behaviour is modeled in the following three generic types:

- **One2One-FP/BP**: one instance triggers another at two different boards, resulting in one instance pair in the way of FP or BP, respectively.
- **One2Many-FP/BP-Static**: one instance triggers other n instances at $n+1$ different boards, resulting in n instance correlations in the way of FP or BP, respectively. The value of n is constant regardless of dynamic traffic distribution.
- **One2Many-FP/BP-Dynamic**: one instance triggers other n instances taking place at $n+1$ different boards, resulting in n instance correlations in the way of FP or BP, respectively. The value of n depends on the traffic distribution.

IV. Proposed IC-FD Approach

Fig. 2 shows the flowchart of the proposed IC-FD approach that aims to construct the ICT(s) according to a set of alarms during an observation window in real time. Given the raw dataset collected from the historical operations, our first step is to obtain attractive features from each instance pair as shown in (i) of Fig. 2. The details of feature extraction are given in IV.A. Then the obtained training dataset is used to train a binary classifier that learns the similarity measure of an instance pair, as shown in (ii) of the flowchart whose details will be given in IV.B.

With the learned similarity measure, given a testing dataset collected by observing a network within an observation window P, all correlations among the instances can be explored

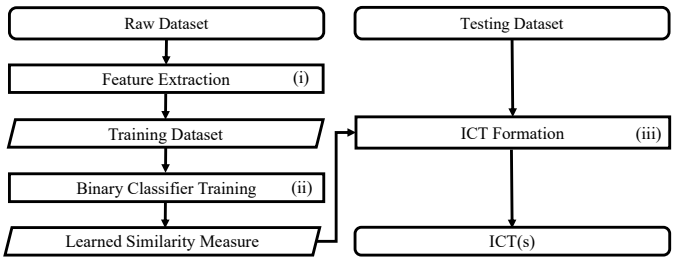


Fig. 2. Flowchart of the proposed IC-FD approach.

through the ICT formation process as illustrated in (iii) of Fig. 2 that will be detailed in IV.C. Eventually, one or a set of ICTs corresponding to the given alarm set shall be obtained as the output of the proposed approach.

A. Feature Extraction

The raw dataset is provided according to historical data from the carrier operators, which is collected within a set of observation windows denoted as $\mathcal{T} = \{T_1, \cdots, T_k, \cdots, T_K\}$, $\forall k \in \{1, \cdots, K\}$. During T_k we observed the network topology G_{T_k}, a set of lightpaths denoted as L_{T_k}. Here, G_{T_k} represents the interconnection of the boards and each board could be a fiber segment or device whose failure would interrupt the traversing optical traffic flows. A number of N_{T_k} alarm instances that have been sorted in ascending order of their occurrence time, are denoted by $A_{T_k} = \{a_1, \cdots, a_{N_{T_k}}\}$. A number of M_{T_k} board instances, denoted as $F_{T_k} = \{f_1, \cdots, f_{M_{T_k}}\}$, are obtained by considering all boards in G_{T_k}. Based on A_{T_k} and F_{T_k}, a set denoted as D_{T_k} with a size far smaller than $\frac{1}{2}[(N_{T_k}+M_{T_k})^2-(N_{T_k}+M_{T_k})]$, contains all possible instance pairs. A set of ground-truth instance correlations denoted as $U_{T_k} = \{f_j \rightarrow a_i | i \in \{1, \cdots, N_{T_k}\}, j \in \{1, \cdots, M_{T_k}\}\} \bigcup \{a_i \rightarrow a_{i'} | i, i' \in \{1, \cdots, N_{T_k}\}\}$, is prepared. It is required in the training process by generating the binary label of each instance pair corresponding to D_{T_k}.

To characterize each instance pair $(v_i, v_j) \in D_{T_k}$, we create a feature space $\mathcal{H}$ in dimensions of time, board/alarm attribute, network topology, and traffic distribution. The feature vector of (v_i, v_j), denoted as $\mathcal{H}(v_i, v_j)$, is defined in (1):

$$\mathcal{H}(v_i, v_j) = [\Delta t_{i,j}, |l_{i,j}|, M(l_{i,j}), b_i, m_i, r_i, b_j, m_j, r_j, C(l_{i,j})], \tag{1}$$

where $\Delta t_{i,j}$ denotes the time gap of two instances, $|l_{i,j}|$ denotes the length of the shortest path from h_i to h_j in G_{T_k}, and $M(l_{i,j})$ counts the number of OMS connections traversed by boards on the shortest path. b_i, m_i, r_i and b_j, m_j, r_j are the board type, instance type, and layer type of instance v_i and v_j, respectively. Finally, $C(l_{i,j})$ is a binary value that indicates whether h_i and h_j are traversed by a common OCH and can be obtained by checking L_{T_k}. Note that for the instance pair that contains a board instance, the missing features are complemented, where the occurrence time is set as the beginning moment of the observation window and an additional layer type is introduced.

By mapping each instance pair from D_{T_k} to $\mathcal{H}$, the transformed dataset $\mathcal{H}(D_{T_k})$ can be obtained. $\forall k \in \{1, \cdots, K\}$, the training dataset derived from all historical data, which is denoted as $D = \bigcup_{k=1}^{K} \mathcal{H}(D_{T_k})$, can be obtained for the subsequent binary classifier training.

B. DNN Architecture

The proposed binary classifier is used to evaluate the correlation of an instance pair in terms of the similarity measure. For this, a deep neural network (DNN) is employed whose architecture is illustrated in Fig. 3, where the dimension of each layer is given in parentheses. According to (1), $\mathcal{H}(v_i, v_j)$ consists of the numerical features including $\Delta t_{i,j}, |l_{i,j}|, M(l_{i,j})$ as well as the categorical features containing $b_i, m_i, r_i, b_j, m_j, r_j, C(l_{i,j})$. To combine these two types of input features, we map the categorical features of $\mathcal{H}(v_i, v_j)$ in continuous space by an embedding layer. Its output is concatenated with the numerical features of $\mathcal{H}(v_i, v_j)$ and fed into the subsequent fully-connected layers. Eventually, the model outputs the probability of instance correlation $v_i \rightarrow v_j$, denoted as $Pr\{v_i \rightarrow v_j\}$. In addition, since most

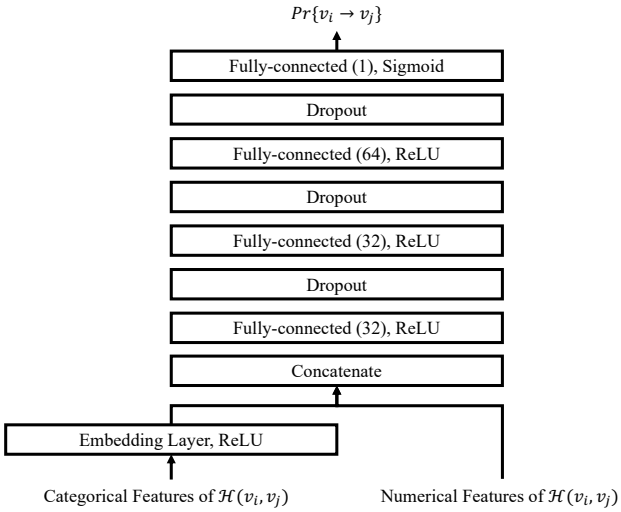


Fig. 3. Architecture of DNN.

instance pairs are non-correlated, D is imbalanced. We adopt the resampling approach to build a balanced dataset through oversampling the minority class by random duplication [26] [27].

C. ICT Formation

During an observation window P, the testing dataset of an arbitrary network state is provided, which incorporates the network topology G_P, a set of lightpaths L_P, and an alarm instance set A_P. The set of board instances is denoted as $F_P = \{f_1, \cdots, f_{M_P}\}$. As shown in Fig. 4, the goal of the proposed ICT formation is to construct one or multiple ICTs, each with a board instance as the root and some alarm instances as the leaf nodes, such that the alarm instances can be covered by the ICT(s) to the maximum extent. With the

ICT(s), the required failure localization and alarm correlation can be achieved.

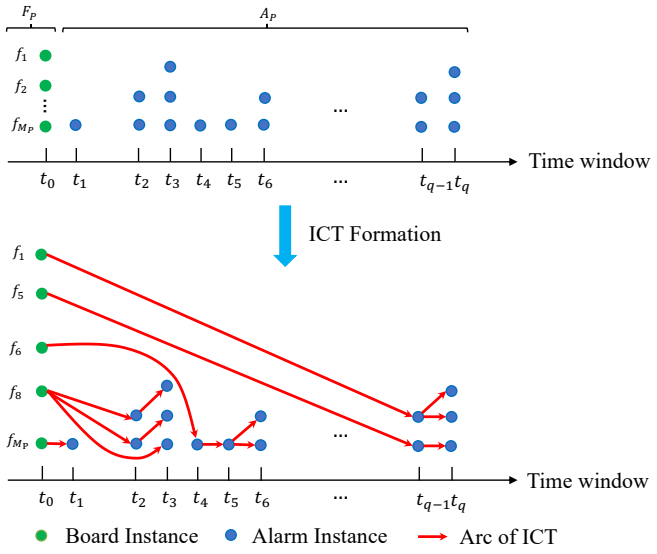


Fig. 4. Illustration of the input and output of ICT formation.

The following assumptions are held in the proposed ICT formation process. Firstly, the state of any board in F_P can only change at most once at the beginning of each observation window, i.e., either staying normal or switching from normal to failed. Once a board is failed, it stays in the failed state until the end of the observation window. This implies that a board instance may not correlate to any alarm instance and will not be taken into the ICT formation process. Secondly, all alarm instances in A_P are caused by one or multiple boards being failed at the starting moment of P. If there exists any alarm instance associated with a board that wasn't faulty at the beginning of P, it could be removed by examining the historical ICTs. Thirdly, due to the characters of failure/alarm propagation behaviour in OTN, *One2One* and *One2Many* could happen whereas *Many2One* isn't allowed.

Based on the above assumptions, one or multiple ICTs constitute a directed forest denoted as $\mathcal{G}_P^I = (V_P^I, E_P^I)$, where V_P^I is the vertex set of instances containing the faulty board instances and all alarm instances, and E_P^I is the arc set covering all identified instance correlations.

To provision the search space for discovering $\mathcal{G}_P^I$, a weighted directed acyclic graph $\mathcal{G}_P$ is defined to incorporate all possible instance correlations, denoted as $\mathcal{G}_P = (V_P, E_P, W(E_P))$. $V_P = F_P \bigcup A_P$ is the vertex set of all board instances and alarm instances. $E_P = E_P^1 \bigcup E_P^2$ is the set of arcs, where $E_P^1 = F_P \times A_P$, $E_P^2 \subseteq A_P \times A_P$ are the sets of all possible instance correlations. The alarm instances in A_P have been sorted in ascending order of their occurrence time. E_P^2 can be obtained by considering all alarm instance pairs whose time gap is greater than 0. Whereas $W(E_P)$ is the set of non-negative weights for arcs in E_P. $\forall v_i, v_j \in V_P, (v_i, v_j) \in E_P, w_{ij} \in W(E_P)$ represents the cost of instance correlation $v_i \to v_j$ and it's denoted in (2):

$$w_{ij} = \begin{cases} 1 - Pr\{v_i \text{ is failed}\} \\ \quad \cdot Pr\{v_i \to v_j\}, \text{if } (v_i, v_j) \in E_P^1, \\ 1 - Pr\{v_i \to v_j\}, \text{if } (v_i, v_j) \in E_P^2, \end{cases} \tag{2}$$

where $Pr\{v_i \text{ is failed}\}$ is the probability that the board instance v_i becomes faulty. It can be determined by the probability density function of time-to-failure (TTF) of the corresponding board, which is estimated according to the historical failure events hitting this board. $Pr\{v_i \to v_j\}$ is the probability of instance correlation $v_i \to v_j$, which is calculated by the trained binary classifier.

1) Integer Linear Programming (ILP): The problem of abstracting the best possible $\mathcal{G}_P^I$ from $\mathcal{G}_P$ can be formulated as an ILP given as follows:

$$\text{minimize} \quad \sum_{e \in E_P} w_e x_e + \sum_{u \in F_P} y_u \tag{3a}$$

$$\text{subject to} \quad \sum_{e \in \delta^-(u)} x_e = 1, \quad \forall u \in A_P, \tag{3b}$$

$$x_e \le y_u, \quad \forall u \in F_P, \forall e \in \delta^+(u), \tag{3c}$$

$$x_e \in \{0,1\}, \quad \forall e \in E_P, \tag{3d}$$

$$y_u \in \{0,1\}, \quad \forall u \in F_P, \tag{3e}$$

where $\forall e \in E_P, w_e \in W(E_P), \forall u \in V_P, \delta^-(u), \delta^+(u)$ are the sets of all incoming arcs and outgoing arcs of vertex u, respectively. Two binary variables x_e, y_u are defined, where x_e takes 1 if the instance correlation arc e is chosen by $\mathcal{G}_P^I$ and 0 otherwise; while y_u takes 1 if the board instance vertex u is selected as a tree root in $\mathcal{G}_P^I$ and 0 otherwise. The objective function (3a) aims to find the $\mathcal{G}_P^I$ that minimizes the total cost of selected instance correlations and board instances. Constraint (3b) indicates that for each alarm instance vertex, only one incoming arc is selected by $\mathcal{G}_P^I$. This guarantees that all alarm instances are traversed by $\mathcal{G}_P^I$ and the in-degree of each alarm instance vertex must be one, which satisfies $\mathcal{G}_P^I$'s property of being a directed forest. Constraint (3c) implies that for each board instance, if any one of its outgoing arcs is selected then this board instance vertex must be chosen as a tree root.

By solving the above ILP, the anticipated ICT(s) $\mathcal{G}_P^I$ can be obtained. The root nodes in $\mathcal{G}_P^I$ are faulty boards in the observed network state, where the failure localization is accomplished by checking their location information. Also, $\mathcal{G}_P^I$ elaborates all correlations among the faulty boards and alarms.

Obviously, solving the above ILP model could be subject to intolerably long computation time. Thus, a heuristic scheme is developed to come up with feasible solutions.

2) Heuristic Algorithm: Fig. 5 demonstrates the flowchart of the proposed heuristic algorithm that aims to construct the feasible ICT(s) $\mathcal{G}_P^I$. Given the set of instances V_P and its corresponding instance pair set E_P, we can obtain the set of arc weights $W(E_P)$ by passing through each instance pair from E_P into the pre-trained binary classifier. For simplicity,

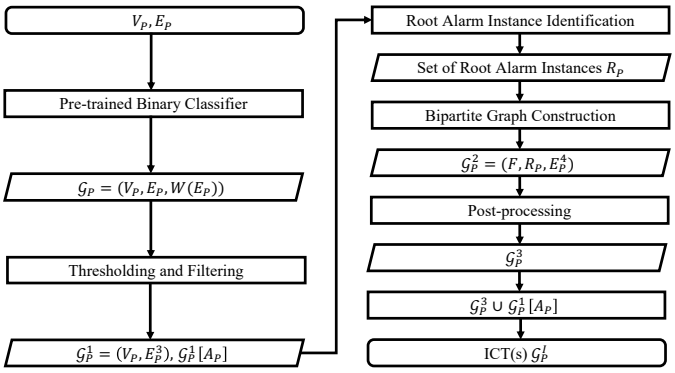


Fig. 5. Flowchart of the proposed heuristic algorithm for ICT formation.

we assume that each board instance v_i is equally likely to be faulty, which implies that $Pr\{v_i \text{ is failed}\}$ can be ignored. To remove unreliable instance correlations in $\mathcal{G}_P$, the arcs whose weight in $W(E_P)$ is greater than 0.5 are discarded. Also, due to the characters of alarm propagation behaviour, for each alarm instance there is at most one incoming arc whose tail vertex belongs to the alarm instance, where we only reserve one arc with the minimum weight if there are multiple qualified incoming arcs. Hence, $\mathcal{G}_P$ is reduced to be $\mathcal{G}_P^1 = (V_P, E_P^3)$. Meanwhile, the subgraph of $\mathcal{G}_P^1$ induced by A_P has become a directed forest, which represents all instance correlations formed by alarm instances and it's denoted as $\mathcal{G}_P^1[A_P]$.

Furthermore, the set of root alarm instances, denoted as $R_P \subseteq A_P$, can be identified as the alarm instance whose in-degree in $\mathcal{G}_P^1$ is non-zero and its all incoming arcs are initiated by board instances. For each root alarm instance $r_i \in R_P$, the corresponding set of board instances F_i and the set of associated alarm instances A_i can be determined by $\mathcal{G}_P^1$, where F_i contains all board instances that connect to r_i and A_i aggregates all alarm instances that are reachable from r_i. Based on the attributes of r_i and all alarm instances in A_i, the board instance whose confidence not only surpasses 0.5 but also is the highest one could be chosen according to the mined association rules between the faulty board and its corresponding alarm types in the raw dataset and all other board instances are eliminated from F_i. In addition, the set of board instances F is acquired by taking the union of all F_i's.

To reflect the relationship between the set of board instances and root alarm instances, we define a directed bipartite graph $\mathcal{G}_P^2 = (F, R_P, E_P^4)$, where $E_P^4 \subseteq E_P^3$ is the arc set that represents all instance correlations between the board instances in F and root alarm instances in R_P. We will post-process it and denote the output as $\mathcal{G}_P^3$, which aims to cover the root alarm instances to the maximum extent by choosing the least number of board instances as the faulty boards. For each component in $\mathcal{G}_P^2$, we iteratively select one board instance with the maximum out-degree and all of its outgoing arcs until all root alarm instances have been explained by the corresponding board instances.

Eventually, the feasible ICT(s) $\mathcal{G}_P^I$ shall be obtained by taking the union of $\mathcal{G}_P^3$ and $\mathcal{G}_P^1[A_P]$. Note that there could exist more than one $\mathcal{G}_P^3$ for the same $\mathcal{G}_P^2$, leading to multiple $\mathcal{G}_P^I$'s, where we will take the union of all those solutions in case missing any possible faulty board.

V. Case Study

Extensive case studies are conducted to verify the proposed IC-FD method in OTN and compare it with a number of counterparts. An OTN simulator [29] is firstly developed to generate ground-truth alarms and the resultant ICTs $\mathcal{G}_P^T$ according to the given failure event, failure/alarm propagation rule database, as well as G_P and L_P during the observation window P. The ICTs produced by the proposed IC-FD approach are denoted as $\mathcal{G}_P^E$. Currently, the relational rule database contains 39 rules, which incorporates 65 instance correlation types formed by 20 failure types, 26 alarm types, and 16 board types. Without loss of generality, each failure event will independently hit a board and thus affecting the traversing optical flows.

The goal of this case study includes the following two aspects:

1) evaluate the performance of the trained binary classifier on the training/validation set.
2) verify the generality and migratability of the proposed IC-FD by comparing its performance with that of the counterparts on the testing datasets of single board failure in the following network environments:
   - use the same network topology and lightpath setting as the raw dataset;
   - change the number of lightpaths for the given network topology;
   - change the size of network topology for the given number of lightpaths;
   - change the ratio of the number of noise alarms to that of true alarms.

The state-of-the-art counterparts considered in this case study include BP [19], LSTM [19], GNN [22]–[24], Transformer [25], and CNN [30].

A. Setup

1) Raw Dataset: We generate the raw dataset by initiating a set of independent single board failure events, where each of them in turn hits one board in the given network topology. The length of each observation window T_i is 1 min. The network topology G_{T_i} is characterized by S_i, $|F_{T_i}|$, deg_i, which are the number of nodes, the number of board instances, and the board-level average degree, respectively. Whereas the set of lightpaths L_{T_i} is described by $|L_{T_i}|$ and $|\overline{l_i}|$ that indicate the number of lightpaths and the average number of boards traversed by each lightpath. The setting of network topology and lightpath are consistent in all observation windows, where $\forall i \in [1, 561], S_i = 15, |F_{T_i}| = 561, deg_i = 2.48, |L_{T_i}| = 40$, and $|\overline{l_i}| = 14$.

2) Training Dataset: The raw dataset contains 7365 alarms, leading to a training dataset of size 2726247, where the ratio of positive samples to negative samples is 4970:2721277. We set a 64%, 16%, and 20% split for training, validation and test sets. For the numerical features, min-max scaling is applied for normalization. We adopt the binary cross-entropy loss function, which is optimized with Adam at a learning rate of 0.001. The batch size and the number of epochs are set to 450 and 100. Also, the technique of early stopping [28] is applied to reduce overfitting, which monitors the value of the area under the curve (AUC) on the validation set in each epoch.

3) AI Architectures of the Counterparts: The network architectures of the counterparts are briefly described as follows. For BP and LSTM, two networks of $233 \times 64 \times 32 \times 561$ and $233 \times 64 \times 561$ are constructed. For training the GNN, an alarm knowledge graph with 43 entity nodes is built based on the failure/alarm propagation rule database. The CNN model consists of 24 input layer units as well as 3 hidden layers whose number of neurons are 256, 128, and 32, where the kernel size in each hidden layer is 2×2. While the Transformer encodes the alarm context with a 768-dimension vector and sets the length of each alarm transaction to 5.

B. Performance Metrics

The comparison between $\mathcal{G}_P^E$ and $\mathcal{G}_P^T$ is accomplished via three parts.

1) Metrics for Root Alarm Identification: Firstly, we evaluate the results of identified root alarms in terms of *precision(R)*, *recall(R)*, and *accuracy(R)*, which are defined as follows:

$$precision(R) = \frac{NC_{ra,E}}{NT_{ra,E}}, recall(R) = \frac{NC_{ra,E}}{NT_{ra,T}}, accuracy(R) = \frac{NC_{a,E}}{|A_P|}, \tag{4}$$

where $NC_{a,E}$ is the number of correctly inferred root and non-root alarm instances according to $\mathcal{G}_P^E$; $|A_P|$ is the size of alarm instance set A_P; $NC_{ra,E}$ is the number of correctly inferred root alarm instances in $\mathcal{G}_P^E$; $NT_{ra,E}$, $NT_{ra,T}$ are the number of root alarm instances in $\mathcal{G}_P^E$ and $\mathcal{G}_P^T$, respectively.

2) Metrics for Failed Board Identification: Secondly, the performance of failure localization in terms of how precisely/accurately the failed boards can be identified is assessed by comparing the faulty boards in $\mathcal{G}_P^E$ with that in $\mathcal{G}_P^T$, which is evaluated via *precision(F)*, *recall(F)*, *accuracy(F)*, as given by:

$$precision(F) = \frac{NC_{root,E}}{NT_{root,E}}, recall(F) = \frac{NC_{root,E}}{NT_{root,T}}, accuracy(F) = \frac{NC_F}{|F_P|}, \tag{5}$$

where NC_F is the number of correctly inferred functioning and faulty boards according to $\mathcal{G}_P^E$; $|F_P|$ is the total number of boards in G_P; $NC_{root,E}$ is the number of correctly inferred faulty boards in $\mathcal{G}_P^E$; $NT_{root,E}, NT_{root,T}$ are the number of faulty boards in $\mathcal{G}_P^E$ and $\mathcal{G}_P^T$, respectively.

3) Metrics for Alarm Instance Correlation: Thirdly, we evaluate the quality of alarm instance correlations in ICTs in terms of *recall(A)* and *accuracy(A)*, which are defined by:

$$recall(A) = \frac{NC_{arc,E}}{NT_{arc,T}}, accuracy(A) = \frac{NC_A}{|D_P^A|}, \tag{6}$$

where NC_A is the number of correctly labeled alarm instance pairs according to $\mathcal{G}_P^E$; $|D_P^A|$ is the size of alarm instance pair set D_P^A; $NC_{arc,E}$ is the number of correctly inferred alarm instance correlations in $\mathcal{G}_P^E$ and $NT_{arc,T}$ is the number of alarm instance correlations in $\mathcal{G}_P^T$.

C. Results

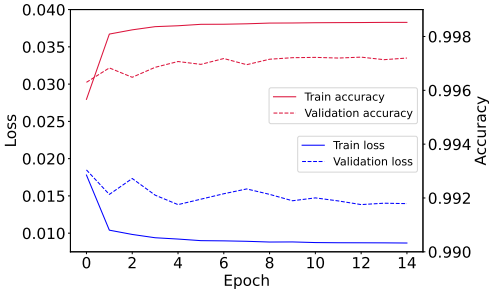


Fig. 6. DNN performance on the training and validation set.

1) Binary Classifier: The training procedure of DNN is demonstrated in Fig. 6. The loss converged to 0.0087/0.014 and accuracy reached 99.85%/99.72% after 15 epochs on the training/validation set.

TABLE I
PERFORMANCE COMPARISON AMONG DIFFERENT SCHEMES UNDER THE NETWORK ENVIRONMENT OF THE RAW DATASET

Schemes \ Metrics	# of Trainable Parameters	$precision(R)$	$recall(R)$	$accuracy(R)$
IC-FD	5585	1	0.9666	0.9975
CNN	150531	0.895	0.9666	0.9627
Transformer	2521716	0.2153	0.2666	0.7156
		$precision(F)$	$recall(F)$	$accuracy(F)$
IC-FD	5585	0.95	1	0.9998
BP	35569	0.85	0.9	0.9994
LSTM	112753	0.85	1	0.9994
GNN	51999	0.5833	0.7	0.7

2) ICT Formation: Firstly, we adopted the network setting of the raw dataset and conducted 10 independent single board failure experiments. The average performance results of various schemes are summarized in Table I. Apparently, the IC-FD achieved a significant advantage in terms of all metrics regarding root alarm instance/failed board identification against the counterparts at the cost of the least number of parameters.

Furthermore, we verified the migratability of IC-FD where the model was trained by using raw data from an initial setting while being applied to some other network scenarios with different topologies and traffic distribution. On one hand, given a network topology whose $S_i = 14, |F_{P_i}| = 3470, deg_i = 2.08$, we varied the number of lightpaths $|L_{P_i}|$ from 50 to 500. On the other hand, we fixed $|L_{P_i}| = 200, |\overline{l_i}| = 16$ and changed the size of network topology, where $S_i \in [10, 37], |F_{P_i}| \in [1074, 1839], deg_i \in [2.1, 2.32]$. As shown in Fig. 7(a)(b), the IC-FD performed the best in root alarm identification among those three methods, where its $accuracy(R)$, $recall(R)$ stabilized above 97%, 90% and its $precision(R)$ remained above 93%. However, the performance by the CNN model was

significantly degraded under certain values of $|L_{P_i}|/|F_{P_i}|$ and that by Transformer behaved even worse, which shows that they can't stably capture root alarms when the spatial relation and traffic distribution were completely ignored. Further as depicted in Fig. 7(d)(e), $recall(F)$ of IC-FD maintains 1 that implies no true faulty boards were missed, which wasn't accomplished by LSTM and BP even if their $accuracy(F)$ and $precision(F)$ are similar to that of IC-FD. Whereas the performance of GNN exhibited high fluctuation due to merely learning the mapping between the failure type and alarm type. Also, as displayed in Fig. 7(g)(h), most alarm instance correlations were successfully identified by IC-FD. Note that since the location information of faulty board/alarm varies with different network environments, all AI models taken by the counterparts need to be retrained as long as there is any change with the network topology/traffic distribution, whereas the IC-FD only needs to be trained once with the alarm data collected from any given network state(s) but it showed the best migratability among all counterparts.

Finally, we tested the anti-noise capability of IC-FD by introducing some noise alarms on top of the true alarms due to a single board failure event. As shown in Fig. 7(c), CNN and Transformer suffered serious performance degradation in detecting root alarms as the ratio of noise alarms continues to increase, whereas IC-FD demonstrated a good capability in overcoming the noises thanks to its additional consideration of the spatial relations among the received alarms. Similarly as illustrated in Fig. 7(f), IC-FD can keep steady performance in faulty board identification when encountering the noises while all the counterparts are subject to significant degradation.

Note that the total processing time of IC-FD is proportional to the number of examined instance pairs each taking about 4 ms to handle.

VI. Conclusion

This paper introduced a novel failure localization and alarm analysis scheme in the optical layer of OTN, called Instance Correlation based Fault Diagnosis (IC-FD), aiming to identify the affected boards due to a failure event with high precision in real time. The proposed IC-FD is characterized by a suite of novel modeling approaches. Firstly, we have included both the alarm instances and board instances in the correlation model so as to perform localization of the failed boards directly instead of merely obtaining the root alarms. Secondly, we developed a novel DNN-based binary classifier along with various features that consider all those static and dynamic network parameters, aiming to achieve sufficient generality and migratability for various network environments. Thirdly, the ICT can effectively describe the correlation between the alarms and the faulty boards, and the ICT formation process serves as a graceful solution that can swiftly come up with high-quality results.

Extensive case studies were conducted to verify the feasibility and performance of the proposed method and modeling approaches. Compared with its counterparts, the proposed IC-FD scheme can adapt to versatile network environments (i.e., change of network topology, traffic distribution, or adding noise alarms) and achieve superb and stable performance in root alarm and failed board identification in terms of precision, recall, and accuracy.

References

[1] International Telecommunications Union-Telecommunication, *G.709:Interfaces for the optical transport network,* [Online]. Available: https://www.itu.int/rec/T-REC-G.709-202006-I/en2020.

[2] G. Jakobson and M. Weissman, "Alarm correlation," *IEEE Network*, vol. 7, no. 6, pp. 52–59, Nov. 1993, DOI: 10.1109/65.244794.

[3] G. Jakobson and M. Weissman, "Real-time telecommunication network management: Extending event correlation with temporal constraints," in *Proceedings of the Fourth International Symposium on Integrated Network Management*, Boston, MA, USA, 1995, pp. 290–301.

[4] J.-G. Lou, Q. Fu, Y. Wang, and J. Li, "Mining dependency in distributed systems through unstructured logs analysis," *ACM SIGOPS Operating Systems Review*, vol. 44, no. 1, pp. 91–96, 2010.

[5] L. Abele, M. Anic, T. Gutmann, J. Folmer, M. Kleinsteuber, and B. Vogel-Heuser, "Combining knowledge modeling and machine learning for alarm root cause analysis," *IFAC Proceedings Volumes*, vol. 46, no. 9, pp. 1843–1848, 2013.

[6] M. Klemettinen, H. Mannila, and H. Toivonen, "Rule discovery in telecommunication alarm data," *Journal of Network and Systems Management*, vol. 7, no. 4, pp. 395–423, 1999.

[7] K. Hatonen, M. Klemettinen, H. Mannila, P. Ronkainen, and H. Toivonen, "TASA: Telecommunication alarm sequence analyzer or how to enjoy faults in your network," in *Proceedings of NOMS '96 - IEEE Network Operations and Management Symposium*, Kyoto, Japan, 1996, pp. 520–529.

[8] J. Wang, C. He, Y. Liu, G. Tian, I. Peng, J. Xing, X. Ruan, H. Xie, and F. L. Wang, "Efficient alarm behavior analytics for telecom networks," *Information Sciences*, vol. 402, pp. 1–14, 2017.

[9] C. Su, Z. Hailong, and X. Junbiao, "Association mining analysis of alarm root-causes in power system with topological constraints," in *Proceedings of the 2017 International Conference on Information Technology*, Singapore, 2017, pp. 461–468.

[10] H. Mannila, H. Toivonen, and A. Inkeri Verkamo, "Discovery of frequent episodes in event sequences," *Data Mining and Knowledge Discovery*, vol. 1, no. 3, pp. 259–289, 1997.

[11] X. Ao, H. Shi, J. Wang, L. Zuo, H. Li, and Q. He, "Large-scale frequent episode mining from complex event sequences with hierarchies," *ACM Transactions on Intelligent Systems and Technology*, vol. 10, no. 4, pp. 1–26, 2019.

[12] S. Hou and X. Zhang, "Alarms association rules based on sequential pattern mining algorithm," in *2008 Fifth International Conference on Fuzzy Systems and Knowledge Discovery*, Jinan, China, 2008, pp. 556–560.

[13] T. Truong, H. Duong, B. Le, and P. Fournier-Viger, "FMaxCloHUSM: An efficient algorithm for mining frequent closed and maximal high utility sequences," *Engineering Applications of Artificial Intelligence*, vol. 85, pp. 1–20, 2019.

[14] L. Lou, M. Zhang, D. Wang, J. Li, X. Tang, and L. Ai, "Alarm compression based on machine learning and association rules mining in optical networks," in *2018 23rd Opto-Electronics and Communications Conference (OECC)*, Jeju, South Korea, 2018, pp. 1–2.

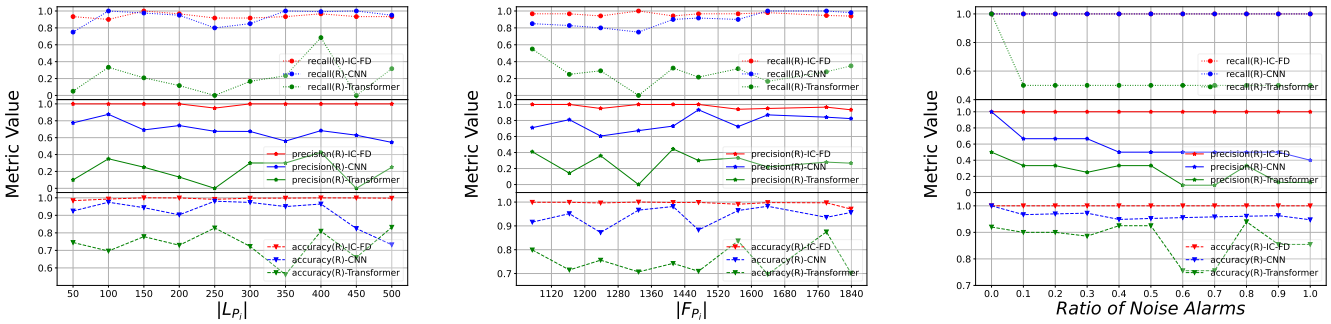
[15] D. Wang, L. Lou, M. Zhang, A. C. Boucouvalas, C. Zhang, and X. Huang, "Dealing with alarms in optical networks using an intelligent system," *IEEE Access*, vol. 7, pp. 97760–97770, 2019.

[16] P. Fournier-Viger, G. He, M. Zhou, M. Nouioua, and J. Liu, "Discovering alarm correlation rules for network fault management," in *International Conference on Service-Oriented Computing*, Dubai, UAE, 2020, pp. 228–239.

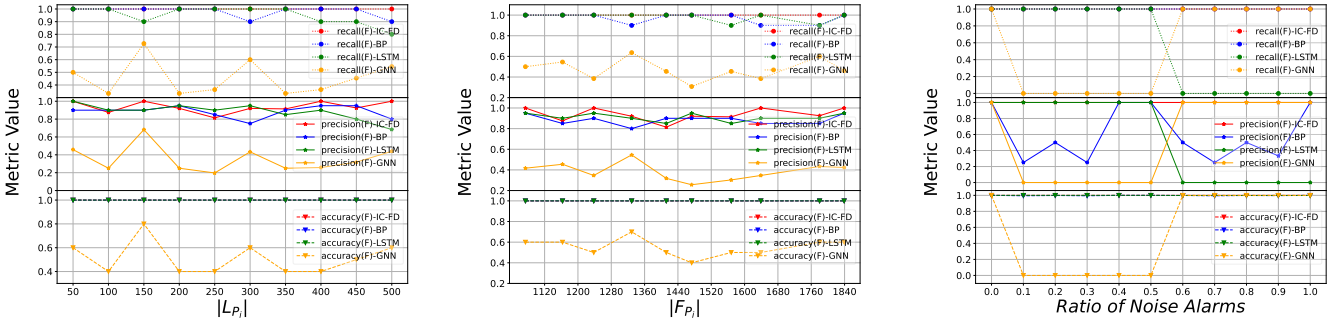
[17] R. Cai, S. Wu, J. Qiao, Z. Hao, K. Zhang, and X. Zhang, "THPs: Topological Hawkes Processes for Learning Causal Structure on Event Sequences," *IEEE Transactions on Neural Networks and Learning Systems*, pp. 1–15, 2022.

[18] 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.

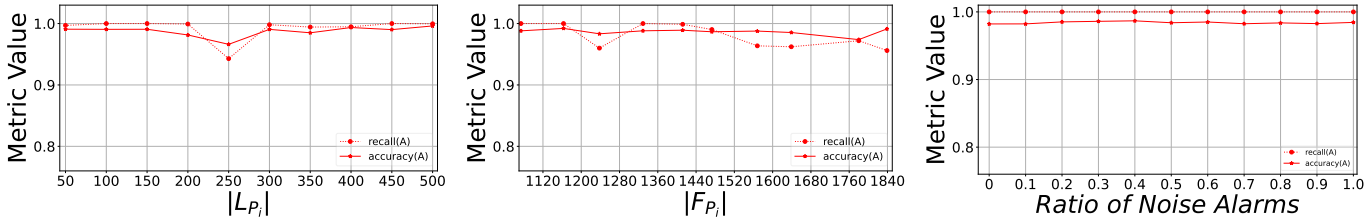
[19] T. Liu, H. Mei, Q. Sun, and H. Zhou, "Application of neural network in fault location of optical transport network," *China Communications*, vol. 16, no. 10, pp. 214–225, 2019.



(a) Root alarm metrics when changing $|L_{P_i}|$ (b) Root alarm metrics when changing $|F_{P_i}|$ (c) Root alarm metrics when adding noise alarms



(d) Failed board metrics when changing $|L_{P_i}|$ (e) Failed board metrics when changing $|F_{P_i}|$ (f) Failed board metrics when adding noise alarms



(g) Alarm instance correlation metrics when changing $|L_{P_i}|$ (h) Alarm instance correlation metrics when changing $|F_{P_i}|$ (i) Alarm instance correlation metrics when adding noise alarms

Fig. 7. Performance comparison of IC-FD and counterparts when changing the network traffic distribution/topology/ratio of noise alarms.

[20] 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, 2019, pp. 1–3.

[21] 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.

[22] 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, 2020, pp. 1–3.

[23] 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, 2020, pp. 1–3.

[24] 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.

[25] 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, 2021, pp. 1–4.

[26] E. Lin, Q. Chen, and X. Qi, "Deep reinforcement learning for imbalanced classification," *Applied Intelligence*, vol. 50, no. 8, pp. 2488–2502, 2020.

[27] TensorFlow, *Classification on imbalanced data:Tensorflow Core,* [Online]. Available: https://www.tensorflow.org/tutorials/structured_data/imbalanced_data.

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

[29] Z. Li, "Design of an OTN-based Failure/Alarm Propagation Simulator," M.S. thesis, Electrical and Computer Engineering, University of Waterloo, Canada, May 2022. [Online]. Available: http://hdl.handle.net/10012/18304.

[30] 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.