ICASSP-SPGC 2022:
ROOT CAUSE ANALYSIS FOR WIRELESS NETWORK FAULT LOCALIZATION

Tianjian Zhang[1,2], *Qian Chen*[1,2], *Yi Jiang*[2], *Dandan Miao*[3],
Feng Yin[1,2], *Tao Quan*[3], *Qingjiang Shi*[1], *Zhi-Quan Luo*[1,2]

[1]Shenzhen Research Institute of Big Data
[2]School of Science and Engineering, The Chinese University of Hong Kong, Shenzhen
[3]Huawei Technologies Co., Ltd

ABSTRACT

Localizing the root cause of network faults is crucial to network operation and maintenance (O&M). Significant operational expenses will be saved if the root cause can be identified agilely and accurately. However, this is challenging for human beings due to the complicated wireless environments and network architectures. Resorting to data analysis and machine learning is promising but remains difficult due to various practical issues, such as the lack of well-labeled samples, hybrid fault behaviors, missing data, and so on. In this paper, we introduce a novel real-world dataset for wireless communication network fault diagnosis. The goal is to infer the root cause timely when we observe certain symptoms in a network. Several baseline methods are provided.

Index Terms— Root Cause Analysis, 5G, Wireless Network

1. INTRODUCTION

Root cause localization of faults in mobile networks is an important task of network operation and maintenance (O&M). By accurately and quickly determining the root causes of network faults, the engineers can take actions timely to repair the core problems. However, the actual network is often involved in a complex wireless communication environment and network deployment structure. Thus, identifying the root cause is rather challenging and with much uncertainty.

In this challenge, we consider a user-concerning problem of slow downlink speed (low value of feature0 in Fig. 1). This could happen in various scenarios, such as crowded stadiums, high-speed car/train, etc. We want to identify the root cause behind timely. The engineers can collect a set of features, key performance indicators (KPIs) in a mobile network that reflect the user network status. The values of these KPIs are time-varying and interacted, following the causal relationship as shown in Fig. 1. Based on these features, we have to infer the root causes of observing abnormally low values of feature0.

The current root cause localization practice in O&M is mainly based on a fault tree [1], which locates the root cause according to the manual rules. While the drawbacks are obvious, for instance, it is hard to generalize to new applications (a new fault tree should be constructed); it is difficult to capture complex relationships, compared to data-driven approaches; the expert knowledge is often limited, etc. How to apply current advanced data-driven approaches for fault localization requires more exploration.

This paper is organized as follows. Section 2 lists the related domains of this signal processing grand challenge (SPGC). Section 3 provides a detailed description of the dataset. Section 4 formulates the problem using mathematical languages. We test several baseline methods in Section 5 and summarize the challenges of this SPGC in Section 6.

2. RELATED TOPICS

Root Cause Analysis (RCA): We focus on the RCA applied to Information Technology (IT) systems. Most RCA models come from the machine learning community, for example, decision tree, support vector machine, neural network, Bayesian network (BN), etc. A comprehensive survey can be found in [2], where considerable models are listed together with their implementation methods and applications.

Time series analysis with missing values: Missing value imputation has been well studied in the past decades, and various algorithms are developed such as single imputation (hot-deck, regression , mean/mode replacement, etc.), multiple imputation (MICE [3], MIDAS [4]). Especially for spatiotemporal data, Chen et al. [5] proposed Bayesian temporal factorization that integrates the low-rank matrix factorization and vector autoregressive process, and provided a detailed comparison across different baselines.

Semi-supervised learning (SSL): High-quality labeled data requires human expertise in O&M practice, leaving lots of collected data samples unlabeled. To address this issue, SSL

This Grand Challenge was organized by the authors and supported by The CUHKSZ-SRIBD-HUAWEI Innovation Laboratory for Future Network Systems Optimization and partly by Guangdong Provincial Key Laboratory of Big Data Computing. The corresponding author is Feng Yin.

978-1-6654-0540-9/22/$31.00 ©2022 IEEE

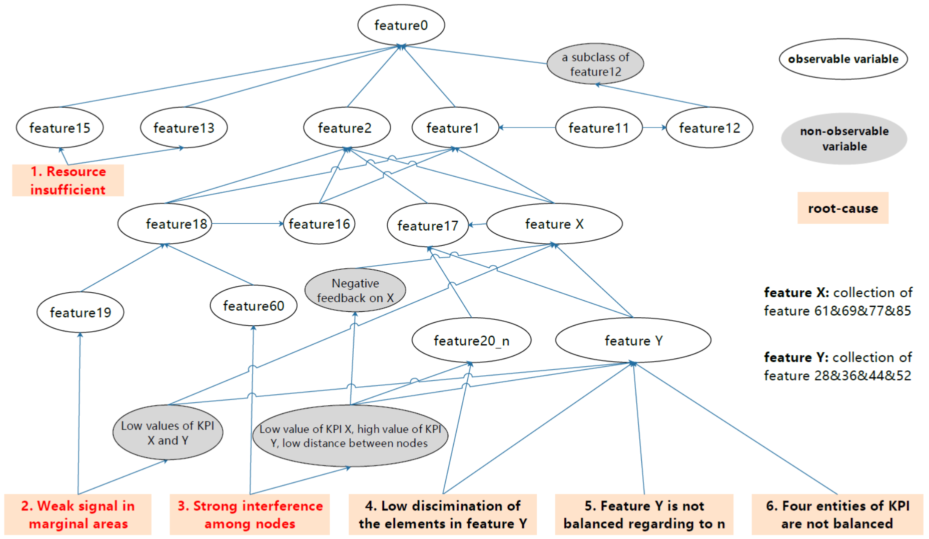
feature0
a subclass of feature12
observable variable
feature15
feature13
feature2
feature1
feature11
feature12
non-observable variable
1. Resource insufficient
root-cause
feature18
feature16
feature17
feature X
feature19
feature60
Negative feedback on X
feature X: collection of feature 61&69&77&85
feature20_n
feature Y
feature Y: collection of feature 28&36&44&52
Low values of KPI X and Y
Low value of KPI X, high value of KPI Y, low distance between nodes
2. Weak signal in marginal areas
3. Strong interference among nodes
4. Low discimination of the elements in feature Y
5. Feature Y is not balanced regarding to n
6. Four entities of KPI are not balanced

Fig. 1. An incomplete causal relationship graph summarized by some domain experts.

can be adopted to make more accurate predictions via exploiting the unlabeled data [6, 7]. For tabular datasets, Wang et al. proposed to combine classifiers optimally for SSL [8]; Guo et al. proposed an SSL algorithm for multi-label cases [9]. For time series classification, SSL methods were also developed in the past two decades, with nearest-neighbor model [10], shapelets learning [11], deep learning model [12], etc.

3. DATA DESCRIPTION

We provide a real-world 5G dataset [1] that includes the following two components.

1. Causal relationship graph (see Fig. 1): This causal relationship graph is a directed acyclic graph (DAG) drawn from a standard communication protocol, which is universal under different scenarios. The oval represents a feature/variable or a set of variables. In particular, a white oval represents an observable variable, and a gray oval represents an intermediate/non-observable variable. Rectangular represents potential root cause. On the top, feature0 is the target variable that the operator cares about. The relationships among different variables are often non-linear. Some of the relationships are deterministic (feature X can be calculated from feature Y), some are probabilistic.

2. Feature dataset: This dataset contains in total 2984 samples. Each sample is a time slice collected from 5G road tests. There are 23 observable variables (white ovals in Fig. 1) carrying information of different KPIs measured within each time slice. Among the 2984 samples, only about 45% of them are carefully labeled with the associated root causes, while the other samples are unlabeled.

3.1. Feature Description

Table 1 shows an example of a time slice from 2020-08-18 18:24:40 to 2020-08-18 18:25:42, where data samples are recorded every second. The values of feature0 stay in a certain range, allowing some jitters. In this example, the values of feature0 are around 300. We label the root cause based on all the data samples in this time slice. Note that multiple root causes may co-exist. For example, the time slice demonstrated in Table 1 is labeled with root cause 2 and 3 simultaneously.

We list the feature description in Table 2, and give the following remarks.

1. Feature values can be continuous or discrete: for example, feature0 is continuous, feature15 is discrete;

2. Some features reveal statistical information: for example, feature3_1∼3_8 represent the counts of feature3 that fall in the corresponding values 1∼8.

3. Spatial characteristics: for example, feature28_0∼28_7 represents feature28's value in direction 0∼7. The values correspond to the receiving signal strength in the spatial direction indicated by feature20_n.

4. Temporal characteristics: for example, the fluctuation of feature19 will result in the unstable behavior of feature0.

5. Different data collection granularity: for example, feature19 is reported every several seconds, while feature0 is recorded every second.

6. Labels are scarce and incomprehensive: in reality, all the labels should be made by engineers, relying on expertise thus expensive. Lots of data remain unlabeled, which may also have faults. In this dataset, we only labeled root causes 1, 2, and 3 (marked in red in Fig. 1); root causes 4, 5, and

[1] Download at https://www.aiops.sribd.cn/

Table 1. Data samples observed in one particular time slice

Time	feature0	feature1	……	feature28_0	feature28_1	……	feature28_7	feature36_0	……
2020-08-18 18:24:40	346.94	24.97	……	-77	-67.81	……	-67.31	-79.5	……
2020-08-18 18:24:41	300.71	25.1	……	-73.94	-67.44	……	-77.56	-73.81	……
……	……	……	……	……					……
2020-08-18 18:25:42	273.64	20.4	……	-60.75	-61.23	……	-70.43	-70.06	……

Table 2. Feature Description

ID	Variable Name	Variable Meaning	Variable Property
0	Date & Time	Timestamp	XXXX-XX-XX XX:XX:XX
1	feature0	Target KPI	Continuous value>0
2	feature1		discrete value: 0∼28
3	feature2		discrete value: normally 0∼4
4	feature3_m, m=1,2,. . . 8	feature2=mean(m*feature3_m)	Continuous non-negative integer
5	feature11		Continuous value: 0∼100
6	feature12		Continuous value: 0∼100
7	feature13		Continuous positive value
8	feature14		Continuous positive value
9	feature15		Continuous positive value
10	feature16		discrete value: 0∼15
11	feature17		discrete value: normally 0∼4
12	feature18		Continuous value
13	feature19		Continuous negative value
14	feature20_n, n=0,1,. . . ,7	ID of 8 receiving directions	Continuous non-negative integer 0-31, arranged as 4*8 matrix: 24,25,26,27,28,29,30,31 16,17,18,19,20,21,22,23 8,9,10,11,12,13,14,15 0,1,2,3,4,5,6,7
15	feature28_n, n=0,1,. . . ,7	Together as a set of feature Y, representing KPI Y; feature28/36/44/52 should be considered jointly regarding direction n.	feature28's strength in direction n, continuous negative value (8 directions in total)
16	feature36_n, n=0,1,. . . ,7		feature36's strength in direction n, continuous negative value (8 directions in total)
17	feature44_n, n=0,1,. . . ,7		feature44's strength in direction n, continuous negative value (8 directions in total)
18	feature52_n, n=0,1,. . . ,7		Feature52's strength in direction n, continuous negative value (8 directions in total)
19	feature60		Continuous negative value
20	feature61_n, n=0,1,. . . ,7	Together as a set of feature X, representing KPI X; Equal to the ratio of feature Y over some other factor; feature61/69/77/85 corresponds to feature 28/36/44/52 respectively.	feature61's strength ratio in direction n, continuous negative value (8 directions in total)
21	feature69_n, n=0,1,. . . ,7		feature69's strength ratio in direction n, continuous negative value (8 directions in total)
22	feature77_n, n=0,1,. . . ,7		feature77's strength ratio in direction n, continuous negative value (8 directions in total)
23	feature85_n, n=0,1,. . . ,7		feature85's strength ratio in direction n, continuous negative value (8 directions in total)

6 remain unlabeled due to the labeling difficulty, while they may also exist in the data.

In reality, we are not able to meet every combination of the faults, while unseen fault combinations may occur at any time. So the label distribution can be very different in the training and test set, see in Fig. 2.

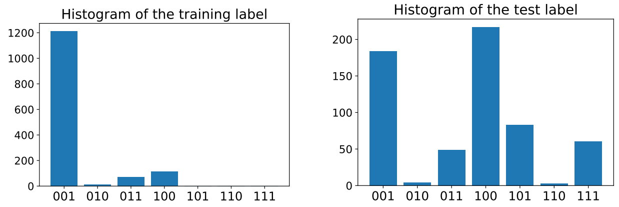

Fig. 2. The training and test label distributions. Horizontal axis shows different root cause combinations, e.g., "101" means the co-occurrence of root causes 1 and 3.

3.2. Performance Evaluation Criterion

We take $N_{te} = 600$ samples to form a test set. The participants should give their judgment on each root cause. We then compare their predictions $\mathbf{p}_i \in \{0,1\}^6$ against our ground truth labels $\mathbf{l}_i \in \{0,1\}^6$, $i = 1, 2, \ldots, 600$.

The evaluation criterion involves the following two steps:

1. Calculate a normalized score s_i for the i-th time slice.
   (1) Initialize $s_i \leftarrow 0$,
   (2) **For** all six root causes, $j \in \{1,2,3,4,5,6\}$:
      a. **If** $p_{ij} = 1$ **and** $l_{ij} = 1$, get 1 mark, $s_i \leftarrow s_i + 1$.
      b. **If** $p_{ij} = 1$ **and** $l_{ij} = 0$, deduct 1 mark, $s_i \leftarrow s_i - 1$.
   (3) Normalize the score, $s_i \leftarrow s_i / \|\mathbf{l}_i\|_1$.
2. Calculate the final score: $S = \sum_{i=1}^{N_{te}} s_i / N_{te}$.

4. PROBLEM FORMULATION

Let us first define the notations. We use capital letters X_i, R_i for random variables and $\mathcal{X}, \mathcal{R}$ for sets of random variables, lowercase $x_i, r_i, \mathbf{x}_j \in \mathbb{R}^d, \mathbf{r}_j \in \{0,1\}^m$ for their realizations, where d and m are the dimension of features and root causes.

Suppose we have a collection of individual systems, each governed by the same set of mechanisms (e.g., physical laws, such as Maxwell's equations), but under different conditions. For example, mobile networks deployed in different cities (wireless communication) or patients with different diseases (healthcare) are such systems. For each individual system j, we can monitor a set of observable variables $\mathcal{X}^{(j)} = \{X_1^{(j)}, \ldots, X_d^{(j)}\}$, to see if the system j works well (normal or abnormal). Note that $X_i^{(j)}$ here could be a random vector of dimension $T^{(j)}$ containing temporal information. When the system malfunctioned due to one or multiple root causes already defined in the set $\mathcal{R}^{(j)} = \{R_1^{(j)}, \ldots, R_m^{(j)}\}$, a subset of $\mathcal{X}^{(j)}$ will behave abnormally. In most situations, we can only observe $\mathcal{X}^{(j)}$ without knowing $\mathcal{R}^{(j)}$ simultaneously. If we can identify the root cause of such symptoms accurately and efficiently, then a huge amount of resources will be saved. In the following, we assume $P(\mathcal{X}^{(j)}, \mathcal{R}^{(j)}) = P(\mathcal{X}^{(k)}, \mathcal{R}^{(k)}), \forall j, k$, and omit the superscripts for simplicity.

Our estimand is $P(\mathcal{R}|\mathcal{X} = \mathbf{x})$, the probability of root causes given the observed variables $\mathbf{x} = [x_1, \ldots, x_d]$. We need a good estimation of that, given N_l labeled data $\{\mathbf{x}_j, \mathbf{r}_j\}_{j=1}^{N_l}$ and N_u unlabeled data $\{\mathbf{x}_j\}_{j=N_l+1}^{N_l+N_u}$. In practice, people may want to know the probability for each cause,

$P(R_i|\mathbf{x}) = \sum_{\mathcal{R}_{\backslash i}} P(\mathcal{R}|\mathbf{x})$, where $\mathcal{R}_{\backslash i}$ denotes for the set of all the root-cause random variables except for the i-th one.

In most applications, the relationships between the variables X_i, R_i are unknown or require knowledge from domain experts. In our dataset, the causal relationships of X_i, R_i are given but may be incomplete.

5. BASELINE METHODS AND EVALUATION

We compared the following baseline methods in this section. For detailed setups of the baseline methods, please refer to our website at https://www.aiops.sribd.cn/.

- **k-Nearest Neighbors (k-NN):** k-NN [13] is a non-parametric classification method that classifies a sample by the vote of its k-nearest neighbors.

- **CatBoost:** CatBoost [14] is an open-source implementation of gradient boosting algorithm that can handle categorical variables. It was shown that CatBoost outperforms other gradient boosted decision tree implementations, such as XGBoost [15], LightGBM [16].

- **Bayesian Network Based Method:** Bayesian network (BN) [17] is a probabilistic graphical model which represents the conditional dependencies among variables via a DAG. We tried the simplest Naive-Bayes [18] structure and a slightly more complicated one extracted from Fig 1. It is also possible to conduct SSL with BN [19]. Given a dataset with N_l labeled samples and N_u unlabeled samples, and a parameterized joint distribution as $P(\mathcal{R}, \mathcal{X}; \theta) = P(\mathcal{X}|\mathcal{R}; \theta)P(\mathcal{R}; \theta)$, the log-likelihood function is:

$$L(\theta) = L_l(\theta) + L_u(\theta) + \text{const.},$$

  where $L_u(\theta) = \sum_{j=(N_l+1)}^{N_l+N_u} \log\left[P\left(\mathbf{x}_j; \theta\right)\right]$ and $L_l(\theta) = \sum_{j=1}^{N_l} \log[P(\mathbf{x}_j|\mathbf{r}_j; \theta)P(\mathbf{r}_j; \theta)]$. Thus, we can estimate $\hat{\theta}$ via maximizing the log-likelihood, using gradient descent or expectation-maximization algorithm [20, 21]. In this paper, we discretized the continuous features into 10 categories and simplified the BN structure such that it only includes the root causes and their children. We implemented the BN using `pgmpy` library [22].

- **MLP:** Neural networks are known as universal approximators. We can directly input the relevant features into a multi-layer perception (MLP) with ReLU activation to predict the root causes.

- **CTO & MASS:** CTO [8] is an SSL algorithm for tabular datasets that combining classifiers optimally. MASS [23] is an implementation of SSL for multi-label cases.

5.1. Data Pre-processing

We first summarized each time slice by their column means (ignoring the missing values) and scaled features by min-max normalization. Then we compared the models under three feature selection strategies: 1) predict each root cause using the set of all features, 2) predict each root cause R_i by its own children, and 3) predict root causes by the children of set $\mathcal{R}$.

5.2. Evaluation of the Baseline Methods

We evaluate the above mentioned baseline methods with 600 test data samples, and the results are given in Table 3.

Table 3. Comparison of the baselines under three settings

Baselines	All	Ch(R_i)	Ch($\mathcal{R}$)
k-NN ($k = 1$)	0.74583	0.67167	0.72250
CatBoost	0.79056	0.51972	0.78083
Naive-Bayes	0.70778	0.73111	0.74111
MLP	0.44083	0.55806	0.71056
BN	-	-	0.68472
BN-SSL	-	-	0.65028
MASS (SSL)	0.62250	-	0.61583
CTO (SSL)	0.56555	0.53944	0.76194

It shows that CatBoost is relatively effective, even without feature selection. With suitable feature selection, the performance of the MLP and SSL methods can be enhanced. Still, the SSL methods perform worse than supervised methods in most cases, probably due to the incorrect models [19] (improper selection of features and/or probabilistic graph).

6. CHALLENGES

There are several drawbacks of the current baselines which can be improved from many aspects.

First, in our data pre-processing procedure, we reduce the data in each time slice by the column means. Thus the missing data issue is alleviated somewhat. However, the temporal information remains unexploited.

Second, the power of unlabeled data is not revealed. SSL is theoretically possible in our setting from causal perspective, where the target is to learn $P(\mathcal{R}|\mathcal{X})$, as commented in [24].

Third, automatic feature engineering is challenging in general. Automatically transforming the features into meaningful representations is challenging, especially when the features have complicated characteristics as remarked in Section 3.1.

Finally, model interpretability is fairly important and trustworthy models are desired. Although some models perform well in the prediction, it may be difficult to extract insights from their predictions. Providing a well-performed yet interpretable model can make the operator feel more confident to use the model.

7. REFERENCES

[1] R.-X. Duan and H.-L. Zhou, "A new fault diagnosis method based on fault tree and Bayesian networks," *Energy Procedia*, vol. 17, pp. 1376–1382, 2012.

[2] M. Solé, V. Muntés-Mulero, A. I. Rana, and G. Estrada, "Survey on models and techniques for root-cause analysis," *arXiv preprint arXiv:1701.08546*, 2017.

[3] S. van Buuren and K. Groothuis-Oudshoorn, "mice: Multivariate imputation by chained equations in R," *Journal of Statistical Software*, vol. 45, no. 3, pp. 1–67, 2011.

[4] R. Lall and T. Robinson, "The MIDAS touch: Accurate and scalable missing-data imputation with deep learning," *Political Analysis*, p. 1–18, 2021.

[5] X. Chen and L. Sun, "Bayesian temporal factorization for multidimensional time series prediction," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 1–1, 2021.

[6] X. J. Zhu, "Semi-supervised learning literature survey," 2005.

[7] O. Chapelle, B. Scholkopf, and A. Zien, "Semi-supervised learning," *IEEE Transactions on Neural Networks*, vol. 20, no. 3, pp. 542–542, 2009.

[8] Z. Wang, L. Yang, F. Yin, K. Lin, Q. Shi, and Z.-Q. Luo, "Optimally combining classifiers for semi-supervised learning," *arXiv preprint arXiv:2006.04097*, 2020.

[9] Y. Guo and D. Schuurmans, "Semi-supervised multi-label classification," in *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*. Springer, 2012, pp. 355–370.

[10] L. Wei and E. Keogh, "Semi-supervised time series classification," in *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, 2006, pp. 748–753.

[11] H. Wang, Q. Zhang, J. Wu, S. Pan, and Y. Chen, "Time series feature learning with labeled and unlabeled data," *Pattern Recognition*, vol. 89, pp. 55–66, 2019.

[12] J. Goschenhofer, R. Hvingelby, D. Rügamer, J. Thomas, M. Wagner, and B. Bischl, "Deep semi-supervised learning for time series classification," *arXiv preprint arXiv:2102.03622*, 2021.

[13] E. Fix and J. L. Hodges, "Discriminatory analysis. nonparametric discrimination: Consistency properties," *International Statistical Review/Revue Internationale de Statistique*, vol. 57, no. 3, pp. 238–247, 1989.

[14] A. V. Dorogush, V. Ershov, and A. Gulin, "CatBoost: gradient boosting with categorical features support," *ArXiv*, vol. abs/1810.11363, 2018.

[15] T. Chen and C. Guestrin, "Xgboost: A scalable tree boosting system," in *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*, 2016, pp. 785–794.

[16] G. Ke, Q. Meng, T. Finley, T. Wang, W. Chen, W. Ma, Q. Ye, and T.-Y. Liu, "Lightgbm: A highly efficient gradient boosting decision tree," *Advances in neural information processing systems*, vol. 30, pp. 3146–3154, 2017.

[17] J. Pearl, *Probabilistic reasoning in intelligent systems: networks of plausible inference*, Morgan kaufmann, 1988.

[18] H. Zhang, "The optimality of naive Bayes," *AA*, vol. 1, no. 2, pp. 3, 2004.

[19] I. Cohen, F. G. Cozman, N. Sebe, M. C. Cirelo, and T. S. Huang, "Semisupervised learning of classifiers: Theory, algorithms, and their application to human-computer interaction," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 26, no. 12, pp. 1553–1566, 2004.

[20] A. P. Dempster, N. M. Laird, and D. B. Rubin, "Maximum likelihood from incomplete data via the EM algorithm," *Journal of the Royal Statistical Society: Series B (Methodological)*, vol. 39, no. 1, pp. 1–22, 1977.

[21] Greg C. G. W. and Martin A. T., "A Monte Carlo implementation of the EM algorithm and the poor man's data augmentation algorithms," *Journal of the American Statistical Association*, vol. 85, no. 411, pp. 699–704, 1990.

[22] A. Ankan and A. Panda, "pgmpy: Probabilistic graphical models using python," in *Proceedings of the 14th Python in Science Conference (SCIPY 2015)*. Citeseer, 2015.

[23] T.-N. Pham, V.-Q. Nguyen, D.-T. Dinh, T.-T. Nguyen, and Q.-T. Ha, "MASS: a semi-supervised multi-label classification algorithm with specific features," in *Asian Conference on Intelligent Information and Database Systems*. Springer, 2017, pp. 37–47.

[24] B. Schölkopf, F. Locatello, S. Bauer, N. R. Ke, N. Kalchbrenner, A. Goyal, and Y. Bengio, "Toward causal representation learning," *Proceedings of the IEEE*, vol. 109, no. 5, pp. 612–634, 2021.