Unsupervised Root-Cause Analysis with Transfer Learning for Integrated Systems

Renjian Pan, Xin Li and Krishnendu Chakrabarty
Department of Electrical and Computer Engineering, Duke University, Durham, NC
{renjian.pan, xinli.ece, krish}@duke.edu

2021 IEEE 39th VLSI Test Symposium (VTS) | 978-1-6654-1949-9/20/$31.00 ©2021 IEEE | DOI: 10.1109/VTS50974.2021.9441030

***Abstract*— The increasing complexity of integrated systems has exacerbated the problems associated with root-cause analysis. Leveraging advances artificial intelligence, a large amount of intelligent root-cause-analysis methods have been proposed in recent years. However, most of these methods rely on root-cause labels from repair history for defective samples, which are often expensive to obtain. In this paper, we propose an unsupervised root-cause-analysis method that utilizes transfer learning. A two-stage clustering method is first developed by exploiting model selection based on the concept of Silhouette score. Next, a data-selection method based on ensemble learning is proposed to transfer valuable information from a source product to improve the root-cause-analysis accuracy on the target product with insufficient data. Two case studies based on industry designs demonstrate that the proposed approach significantly outperforms other state-of-the-art unsupervised root-cause-analysis methods.**

I. Introduction

The increasing complexity of integrated systems has exacerbated the problems associated with system diagnosis. It is extremely difficult to pinpoint the root causes of system failures using manual diagnosis. There is clearly a pressing need for more intelligent diagnosis methods. Advances in artificial intelligence and machine learning have paved the way for a number of intelligent diagnosis methods for integrated circuits, boards and systems [2]-[7]. These methods can be divided into two categories [2]: (i) supervised root-cause analysis, and (ii) unsupervised root-cause analysis. Supervised root-cause analysis has received much attention during the past decade. Leveraging the historical data with diagnostic labels, supervised-learning models are trained to classify the root-causes for defective samples. Popular supervised-learning methods (e.g., support-vector machines, neural networks, decision trees, etc.) have been successfully applied to board-level and system-level root-cause analysis [3]-[5]. However, supervised root-cause analysis is not effective or applicable in scenarios where labels are either insufficient or inaccurate [7].

To overcome the drawback of supervised methods, unsupervised methods have been proposed with less dependency on expert knowledge [2], [6]. Traditional unsupervised methods cluster the failure samples by analyzing their similarity in test results without requiring diagnostic labels, such that the samples in each cluster share similar root causes. However, these methods often lead to poor performance in practice. Without knowing the diagnostic labels, it is difficult to identify the important test items that are highly correlated to system defects. The diagnosis procedure can be misled by unimportant test items [7]. To address this problem, a two-stage root-cause analysis method has been proposed to cluster failure samples by leveraging the pass/fail information [7], which provides important insights for system defects and can be collected inexpensively.

All the above methods require a sufficient volume of data in order to learn accurate root-cause-analysis models. However, in many practical scenarios (e.g., products in the ramp-up phase [10]), these methods cannot be applied because the data collected on the target product are limited. To enhance diagnostic accuracy, one possible approach is to transfer knowledge from similar products with sufficient data, so that the root-cause-analysis model can be accurately learned with the data from not only the target product, but also the similar products. This idea is known as transfer learning in machine learning [8].

In recent years, transfer learning has been successfully applied to root-cause analysis. A knowledge-transfer method based on root-cause mapping and syndrome mapping has been proposed for board-level functional fault diagnosis [9]. A transfer-learning workflow has been proposed for board-level functional fault identification with domain adaptation based on a similarity measure between different products [10]. However, both these methods focus on supervised root-cause analysis. To the best of our knowledge, the idea of transfer learning has not been explored for unsupervised root-cause analysis.

In this paper, we propose an unsupervised root-cause-analysis method with transfer learning. First, we apply multiple clustering methods (e.g., decision-tree clustering [11], K-means clustering [12], and hierarchical clustering [12]) and a model selection algorithm based on the concept of Silhouette score [13] to improve the two-stage root-cause-analysis model in [7]. Next, to facilitate transfer learning, an ensemble learning method [14] is applied to select the valuable samples from a similar source product. These samples are combined with the limited data collected from the target product to train an accurate root-cause-analysis model for the target product. Using case studies based on two industry designs (high-volume products), we show that the proposed approach is effective for root-cause analysis and it significantly outperforms state-of-the-art methods.

The remainder of this paper is organized as follows. In Section II, we review two-stage unsupervised root-cause analysis. In Section III, we describe several different unsupervised root-cause clustering methods and propose our model selection framework. In Section IV, we propose the

978-1-6654-1949-9/21/$31.00 ©2021 IEEE

transfer-learning algorithm based on ensemble learning. Experimental results on two case studies from industry are presented in Section V to validate the proposed method. Finally, we conclude the paper in Section VI.

II. Background

Unsupervised root-cause analysis can be formulated as the problem shown in Fig. 1. Let $\boldsymbol{S} = \{S_1, S_2, \ldots, S_N\}$ denote a set of samples of systems under diagnosis with the test data $\boldsymbol{D} = \{\boldsymbol{d}_1, \boldsymbol{d}_2, \ldots, \boldsymbol{d}_N\}$, where N is the number of samples and $\boldsymbol{d}_n = \{d_{n,1}, d_{n,2}, \ldots, d_{n,T}\}$ denotes the test data for S_n. The test data are assumed to be numerical throughout this paper. We will consider categorical data as part of future work. Let $\boldsymbol{y}_{PF} = \{y_{PF,1}, y_{PF,2}, \ldots, y_{PF,N}\}$ denote the pass/fail information for the systems under diagnosis, where $y_{PF,n}$ is either 0 or 1, representing whether S_n is defect-free or defective. As shown in Fig. 1, systems under diagnosis with the test data $\boldsymbol{D}$ and the pass/fail information $\boldsymbol{y}_{PF}$ are utilized for unsupervised learning and to build an unsupervised learning model. The root-cause suggestion for systems under diagnosis are made based on this unsupervised learning model.

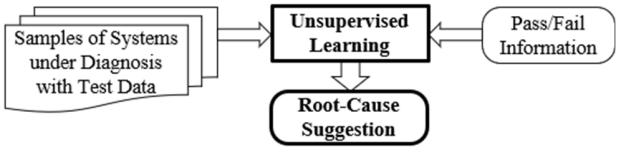

Fig. 1. Problem formulation for unsupervised root-cause analysis.

A two-stage unsupervised root-cause-analysis method has been proposed in [7] and is summarized in Fig. 2. In the first stage, a decision-tree model is built with the test data $\boldsymbol{D}$ and the pass/fail information $\boldsymbol{y}_{PF}$ to roughly cluster the samples. In the second stage, frequent-pattern mining [18] is applied to extract the frequent patterns at each decision-tree node to precisely cluster the samples so that each cluster contains only a few dominant root causes.

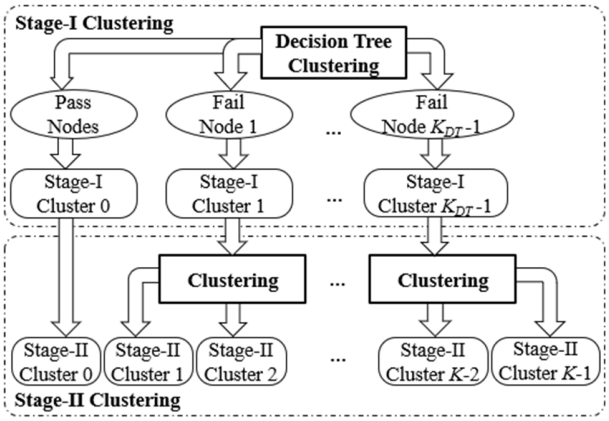

Fig. 2. Two-stage clustering for root-cause analysis.

The decision tree model in the top half of Fig. 2 has been shown to be an effective method to utilize the pass/fail information and select the key features to diagnose whether a sample is defect-free or defective [4]. Based on the test data $\boldsymbol{D}$ labeled with the pass/fail information $\boldsymbol{y}_{PF}$, a tree-based structure is trained with multiple decision paths. Each decision path contains one or more decision rules. The samples in $\boldsymbol{S}$ with the test data $\boldsymbol{D}$ are split into different leaf nodes with different decision paths, and each leaf node naturally forms a cluster.

In the second stage, we apply frequent-pattern mining to further split the Stage-I clusters associated with fail nodes, as shown in the bottom half of Fig. 2. As the samples in $\boldsymbol{S}$ with the test data $\boldsymbol{D}$ pass through the decision tree, we obtain the subsets $\boldsymbol{S}_{I,k}$ and $\boldsymbol{D}_{I,k}$ associated with the k-th Stage-I cluster ($k \geq 1$). For Stage-II clustering on $\boldsymbol{D}_{I,k}$, only the test items ignored by Stage-I clustering are considered since the objective is to extract valuable insights from these unused test items. For each sample $\boldsymbol{s}_{k,n} \in \boldsymbol{S}_{I,k}$, an effective frequent pattern $\boldsymbol{f}_{Eff,k,n}$ is matched with the largest failure rate among all frequent patterns that appear in the sample $\boldsymbol{s}_{k,n}$ [7]. Based on the matched frequent patterns $\{\boldsymbol{f}_{Eff,k,1}, \boldsymbol{f}_{Eff,k,2}, \ldots\}$, the samples in $\boldsymbol{S}_{I,k}$ with the test data $\boldsymbol{D}_{I,k}$ are further split into multiple Stage-II clusters, where samples with the same/different frequent patterns are respectively assigned to the same/different Stage-II clusters.

Although the above method has been shown to be effective for a number of industry test cases [7], it suffers from several limitations. First, only frequent-pattern mining is applied to Stage-II clustering. There are many other clustering methods (e.g., K-means clustering [12], hierarchical clustering [12], etc.) that may outperform frequent-pattern mining for this problem domain. Second, the aforementioned method may not achieve acceptable performance when the test data for the target product are limited. To overcome these limitations, we improve the two-stage clustering method and introduce transfer learning to unsupervised root-cause analysis in the following sections.

III. Two-Stage Clustering

A. Unsupervised Root-Cause Clustering

To improve the performance of the two-stage clustering described in [7], we adopt multiple clustering methods for Stage-II clustering, and propose a model-selection algorithm based on the concept of Silhouette score [13] to select the most effective clustering model. In implementing a new clustering method A_j, we retain the algorithm framework shown in Fig. 2. Instead of only applying frequent-pattern mining, as in [7], the new method A_j is used as the Stage-II-clustering kernel to split each Stage-I cluster into multiple Stage-II clusters, as shown in the bottom half of Fig. 2. Specifically, when we split the subsets $\boldsymbol{S}_{I,k}$ and $\boldsymbol{D}_{I,k}$ associated with the k-th Stage-I cluster, we only consider the test items ignored by Stage-I clustering. A Stage-II-clustering model is built to cluster the samples in $\boldsymbol{S}_{I,k}$ with these test items based on A_j and generate multiple Stage-II clusters.

In addition to frequent-pattern mining, we implement three different clustering methods for Stage-II clustering: (i) K-means clustering [12], (ii) complete-linkage hierarchical clustering [12], and (iii) decision-tree clustering [11]. K-means clustering is one of the most popular clustering methods [12]. Starting from K initial centers of clusters, each sample is assigned to the cluster whose center is closest to it. Next, we recalculate the centers (means) of the K clusters. We repeat this procedure until

convergence is reached. When hierarchical clustering is applied, each sample initially forms a cluster of its own. Next, small clusters are iteratively merged into larger clusters based on their inter-cluster distances. Both K-means clustering and hierarchical clustering have been successfully applied to a variety of testing/diagnosis applications [2], [15]-[17].

The general idea of decision-tree clustering [11] is to convert the clustering problem to a decision-tree classification problem. For each Stage-I cluster in Fig. 2, we conceptually introduce a number of "non-existing" samples. A decision tree model is built to classify the "non-existing" samples and the defective samples. While constructing the decision tree, different Stage-II clusters are generated by different decision paths.

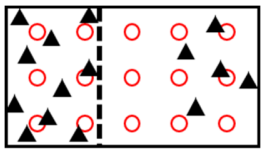

Fig. 3. An example of decision-tree clustering.

We use the simple example in Fig. 3 to illustrate decision-tree clustering. Assume that all the black triangles in Fig. 3 represent the defective samples in one Stage-I cluster. The "non-existing" samples are uniformly added over the feature space as shown by the red circles in Fig. 3. The number of "non-existing" samples is equal to that of defective samples. When we treat the defective samples and the "non-existing" samples as two classes, the clustering problem can be solved by decision-tree classification. Similar to a traditional decision tree, we use information gain as the criterion for partitioning into classes [12]. Information gain represents the reduction of entropy when the data in one decision node is split into two nodes. As shown by the dashed line in Fig. 3, the cutting plane with the maximum information gain is found to split the data. With this cutting plane, two clusters of defective samples are formed. By repeating the above procedure, more clusters can be iteratively generated. The implementation details for decision-tree clustering can be found in [11].

In summary, four clustering models (i.e., frequent-pattern mining, K-means clustering, hierarchical clustering and decision-tree clustering) are implemented for Stage-II clustering. In order to select the most efficient model, we propose a model-selection algorithm as discussed in Section III.C.

B. Hyper-Parameter Selection

The above clustering methods rely on several important hyper-parameters: (i) the number of clusters (i.e., K) for either K-means clustering or hierarchical clustering, and (ii) the minimum number of defective samples in each cluster (i.e., N_{Th}) for either decision-tree clustering or frequent-pattern mining. In this sub-section, we discuss the methodologies to automatically determine these hyper-parameters.

We note that all the hyper-parameters either determine or directly influence the number of clusters in the final clustering result (i.e., K). Considering frequent-pattern mining as an example, if a smaller value of N_{Th} is chosen, more frequent patterns will be identified, and more clusters will be formed. Therefore, we can determine these hyper-parameters by finding the effective value of K.

Based on this observation, we apply the "L-method" to determine all hyper-parameters, following the idea presented in [7]. We generate different clustering models with different hyper-parameter values. Next, we plot the distance metric I_{Dis} as a function of K, as shown in Fig. 4. A "knee point" is found to determine the effective value of K by fitting a piecewise linear function of $I_{Dis}(K)$, which minimizes the mean-squared error [20]. Based on this effective value of K, we find the effective hyper-parameter value.

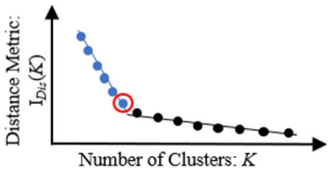

Fig. 4. An example of determining the number of clusters by the L-method.

For K-means clustering, we set the distance metric $I_{Dis}(K)$ as the intra-cluster distance $I_{Intra}(K)$ because it is the cost function associated with K-means clustering and it is one of the most widely used distance metrics for clustering in the literature [12]:

$$I_{Intra}(K) = \left(\sum_{k=1}^{K} \sum_{n=1}^{N_k} \left\| \boldsymbol{d}_{k,n} - \bar{\boldsymbol{d}}_k \right\|_2^2 \right)^{1/2}, \quad (1)$$

where $\|\bullet\|_2$ denotes the L2-norm of a vector, N_k is the number of samples in the k-th cluster, $\boldsymbol{d}_{k,n}$ denotes the test data for the n-th sample in the k-th cluster, and $\bar{\boldsymbol{d}}_k$ is the center of the k-th cluster.

For hierarchical clustering, we use the merged distance $I_{Merge}(K)$ as the distance metric $I_{Dis}(K)$ because it is also the metric used for merging clusters in hierarchical clustering [12]:

$$I_{Merge}(K) = \min_{1 \le i < j \le K+1} \left(\max_{\substack{1 \le m \le N_i \\ 1 \le n \le N_j}} \left\| \boldsymbol{d}_{i,m} - \boldsymbol{d}_{j,n} \right\|_2 \right). \quad (2)$$

It represents the distance when we iteratively merge two nearby clusters and reduce the number of clusters from K+1 to K. For both decision-tree clustering and frequent-pattern mining, we define the distance metric $I_{Dis}(K)$ as the intra-cluster distance $I_{Intra}(K)$ in (1); this definition is similar to what is used in [21]. We determine the effective value of K by the L-method and, subsequently, find the effective value of N_{Th}.

C. Model Selection

After determining all the hyper-parameters and generating all the clustering models, we have to identify which model is most suitable for root-cause analysis with a given product. In order to select the most effective model for Stage-II clustering, we propose a model-selection algorithm based on Silhouette score.

Silhouette score is a metric to evaluate the consistency of data within clusters [13]. It measures the similarity of each sample with respect to its own cluster and the dissimilarity against other

clusters. A higher Silhouette score indicates better clustering performance with most samples assigned to their most matched clusters. Hence, we should choose the most effective clustering model A_{Eff} with the highest Silhouette score among the four candidates (i.e., frequent-pattern mining, K-means clustering, hierarchical clustering and decision-tree clustering).

Algorithm 1 summarizes the overall framework of our proposed unsupervised root-cause analysis with model selection. With the historical test data $\boldsymbol{D}$, a decision-tree model is built to roughly cluster the samples in the first stage as shown in Steps 1-2. After determining the hyper-parameters, we use the Silhouette score to select the most effective model A_{Eff}, as shown in Steps 3-4. Finally, we apply A_{Eff} to Stage-II clustering in Step 5. Each Stage-II cluster represents only a small number of root causes so that human experts can efficiently locate the root causes for samples in each cluster.

Algorithm 1: Unsupervised Root-Cause Analysis

1. Begin with the historical test data $\boldsymbol{D}$.
2. Complete Stage-I clustering with the decision-tree model.
3. Generate four Stage-II clustering models based on frequent-pattern mining, K-means clustering, hierarchical clustering and decision-tree clustering with hyper-parameters determined by the L-method.
4. Select the most effective model A_{Eff} with the highest Silhouette score for Stage-II clustering.
5. Complete Stage-II clustering with A_{Eff} for root-cause analysis.

IV. Transfer Learning

The proposed two-stage clustering method may not achieve the desired level of performance when the volume of historical data for the target product are limited. To address this problem, we propose a transfer-learning algorithm to leverage the prior knowledge from a similar product with sufficient data. We formulate the transfer-learning problem and develop an efficient algorithm based on source data selection by ensemble learning.

A. Problem Formulation

We denote the product under diagnosis as the target product $\mathbf{P}_{Target}$, where $\boldsymbol{S} = \{S_1, S_2, \ldots, S_N\}$ denotes a set of its samples with the test data $\boldsymbol{D}$ and pass/fail information $\boldsymbol{y}_{PF}$. Let $\mathbf{P}_{Source}$ denote the source product, which is similar to $\mathbf{P}_{Target}$ (e.g., the old version of $\mathbf{P}_{Target}$), where $\boldsymbol{S}_{Source} = \{S_{Source,1}, S_{Source,2}, \ldots, S_{Source,M}\}$ denotes a set of its samples with the test data $\boldsymbol{D}_{Source}$ and pass/fail information $\boldsymbol{y}_{PF,Source}$. Note that the same test items are applied to both $\mathbf{P}_{Target}$ and $\mathbf{P}_{Source}$.

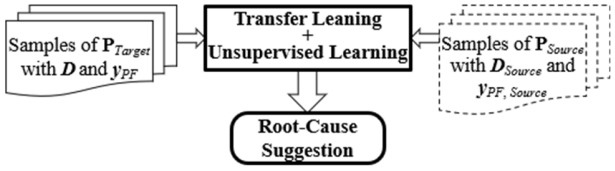

Fig. 5. Problem formulation of unsupervised root-cause analysis with transfer learning.

Unsupervised root-cause analysis with transfer learning can be formulated as shown in Fig. 5. Compared to the formulation in Fig. 1, in which we only use the data (i.e., $\boldsymbol{D}$ and $\boldsymbol{y}_{PF}$) from $\mathbf{P}_{Target}$ for unsupervised learning, we further leverage the data from $\mathbf{P}_{Source}$ (i.e., $\boldsymbol{D}_{Source}$ and $\boldsymbol{y}_{PF,Source}$) by transfer learning. In this way, we can learn a more effective root-cause-analysis model for $\mathbf{P}_{Target}$ with insufficient data.

B. Source Data Selection

To transfer knowledge from $\mathbf{P}_{Source}$, source data selection is an efficient and practical approach. Namely, we select a set of "effective" samples from the source data (i.e., $\boldsymbol{D}_{Source}$ and $\boldsymbol{y}_{PF,Source}$) and combine them with the target data (i.e., $\boldsymbol{D}$ and $\boldsymbol{y}_{PF}$) for model training [14]. Compared to other transfer learning algorithms [22], source data selection is model-free and broadly applicable to a variety of machine learning algorithms. In this paper, we adopt source data selection for our proposed unsupervised root-cause analysis in Algorithm 1.

Without knowing the label information, unsupervised transfer learning is non-trivial [22]. In order to address this challenging problem, we select the source data based upon a supervised method leveraging the pass/fail information. The general framework of source data selection is shown in Fig. 6. In the learning phase, we apply supervised learning to the samples from $\mathbf{P}_{Target}$ and generate a sample selection model. In the selection phase, we choose a set of effective samples from $\mathbf{P}_{Source}$, whose pass/fail information can be accurately predicted by the sample selection model, implying that the statistical characteristics of these selected samples are consistent with those from $\mathbf{P}_{Target}$. Finally, the selected data $\boldsymbol{D}'_{Source}$ and $\boldsymbol{y}'_{PF,Source}$ are merged with $\boldsymbol{D}$ and $\boldsymbol{y}_{PF}$ for root-cause analysis on $\mathbf{P}_{Target}$.

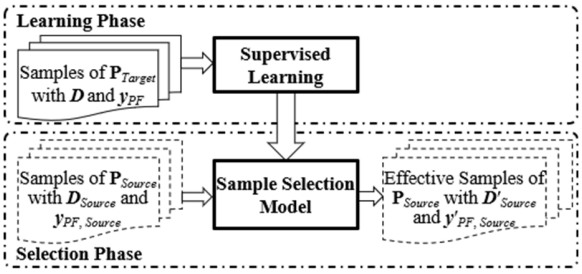

Fig. 6. The proposed framework for source data selection.

To create the sample selection model, we first build K supervised learning models $\boldsymbol{A}_{Boot} = \{A_{Boot,1}, A_{Boot,2}, \ldots, A_{Boot,K}\}$ by using the target data. Each model $A_{Boot,k}$ aims to predict the pass/fail information $y_{PF,n}$ from the test data of a sample $\boldsymbol{d}_n$. It is trained on $\boldsymbol{D}_{Boot,k}$ and $\boldsymbol{y}_{PF,Boot,k}$, which is randomly selected from $\boldsymbol{D}$ and $\boldsymbol{y}_{PF}$ by bootstrapping [12]. Next, for each sample $\boldsymbol{d}_{Source,m}$ from the source data $\boldsymbol{D}_{Source}$, we use all models in $\boldsymbol{A}_{Boot}$ to predict its pass/fail information and calculate the accuracy:

$$P_{Acc,m} = 1 - \frac{\sum_{k=1}^{K} \left| \hat{y}_{PF,m,k} - y_{PF,Source,m} \right|}{K}, \tag{3}$$

where $\hat{y}_{PF,m,k}$ is the prediction of $A_{Boot,k}$ on the sample $d_{Source,m}$. The pass/fail information $y_{PF,Source,m}$ in $\boldsymbol{y}_{PF,Source}$ is used as the reference.

After we evaluate all the samples in the source data (i.e., $\boldsymbol{D}_{Source}$ and $\boldsymbol{y}_{PF,Source}$), we form $\boldsymbol{D}'_{Source}$ and $\boldsymbol{y}'_{PF,Source}$ by selecting all effective samples (i.e., $d_{Source,m}$ and $y_{PF,Source,m}$) with $P_{Acc,m} >$

$P_{Acc,Th}$, where $P_{Acc,Th}$ denotes the threshold for data selection. The data selected in this manner are combined with the target data, and they form the test data $\boldsymbol{D}'$ and the pass/fail information $\boldsymbol{y}'_{PF}$. Finally, two-stage clustering from Algorithm 1 is applied to $\boldsymbol{D}'$ and $\boldsymbol{y}'_{PF}$, and the clustering result associated with the target data is used for root-cause analysis. We summarize the unsupervised root-cause analysis with transfer learning based on source data selection in Algorithm 2.

Algorithm 2: Unsupervised Root-Cause Analysis with Transfer Learning

1. Generate K pairs of training dataset $\{(\boldsymbol{D}_{Boot,1}, \boldsymbol{y}_{PF,Boot,1}), (\boldsymbol{D}_{Boot,2}, \boldsymbol{y}_{PF,Boot,2}), \ldots, (\boldsymbol{D}_{Boot,K}, \boldsymbol{y}_{PF,Boot,K})\}$ by bootstrapping.
2. Learn the model $A_{Boot,k}$ based on $\boldsymbol{D}_{Boot,k}$ and $\boldsymbol{y}_{PF,Boot,k}$ and form the set $\boldsymbol{A}_{Boot} = \{A_{Boot,1}, A_{Boot,2}, \ldots, A_{Boot,K}\}$.
3. Calculate the accuracy $P_{Acc,m}$ in (3) for all samples in the source data.
4. Select the effective source data samples with $P_{Acc,m} > P_{Acc,Th}$ and form $\boldsymbol{D}'_{Source}$ and $\boldsymbol{y}'_{PF,Source}$.
5. Combine the datasets: $\boldsymbol{D}' = \boldsymbol{D} \cup \boldsymbol{D}'_{Source}$ and $\boldsymbol{y}'_{PF} = \boldsymbol{y}_{PF} \cup \boldsymbol{y}'_{PF,Source}$.
6. Apply two-stage clustering in Algorithm 1 to $\boldsymbol{D}'$ and $\boldsymbol{y}'_{PF}$.
7. Use the clustering result associated with the target data $\boldsymbol{D}$ and $\boldsymbol{y}_{PF}$ for root-cause analysis.

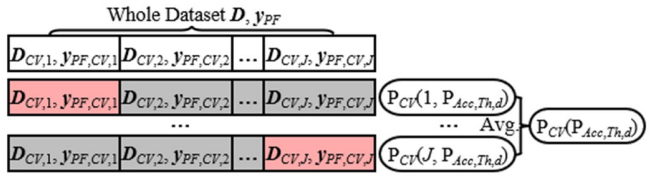

Fig. 7. The framework of cross-validation.

There are two hyper-parameters, namely K and $P_{Acc,Th}$, in Algorithm 2. The value of K is empirically set to 50, which is sufficiently large to ensure that the overall prediction accuracy will not increase with higher K. To determine an effective value of $P_{Acc,Th}$, we apply J-fold cross validation. Let $\{P_{Acc,Th,1}, P_{Acc,Th,2}, \ldots\}$ denote a set of possible values. For each $P_{Acc,Th,d}$, we evenly split the data $\boldsymbol{D}$ and $\boldsymbol{y}_{PF}$ of $\mathbf{P}_{Target}$ into J subsets (i.e., folds) as shown in Fig. 7. Next, for each fold $(\boldsymbol{D}_{CV,j}, \boldsymbol{y}_{PF,CV,j})$ with red color in Fig. 7, we train a decision tree $A_{DT,j}$ to predict the pass/fail information by using the other (J–1)-fold data with grey color in Fig. 7 and the selected source data $(\boldsymbol{D}'_{Source}, \boldsymbol{y}'_{PF,Source})$. The prediction accuracy is denoted as $P_{CV}(j, P_{Acc,Th,d})$. The overall accuracy $P_{CV}(P_{Acc,Th,d})$ is calculated as the arithmetic average of $P_{CV}(j, P_{Acc,Th,d})$. The threshold $P_{Acc,Th,d}$ with maximum $P_{CV}(P_{Acc,Th,d})$ is selected as the effective threshold $P_{Acc,Th}$. In this paper, we set J as 5, which is a typical value [12].

V. Experiments and Results

We demonstrate the effectiveness of the proposed method with two case studies. In both cases, test data from industry are provided for one target product $\mathbf{P}_{Target}$ and one source product $\mathbf{P}_{Source}$ from the product category of network systems. Both $\mathbf{P}_{Target}$ and $\mathbf{P}_{Source}$ are from the same product series and share similarity in their designs. Human experts have manually labeled the root causes of defective samples of $\mathbf{P}_{Target}$ by analyzing available test data and performing additional tests and repairments. The labeled root causes are considered as the ground truth. We perform all experiments on a laptop with 2.6 GHz CPU and 16 GB memory.

TABLE I
Performance of Root-Cause Analysis on Network Product #1

Method	I_{NMI}	K	Runtime (s)
Hierarchical clustering [2]	0.096	30	10.4
Two-stage clustering [7]	0.238	12	11.2
Proposed method	0.290	11	38.6

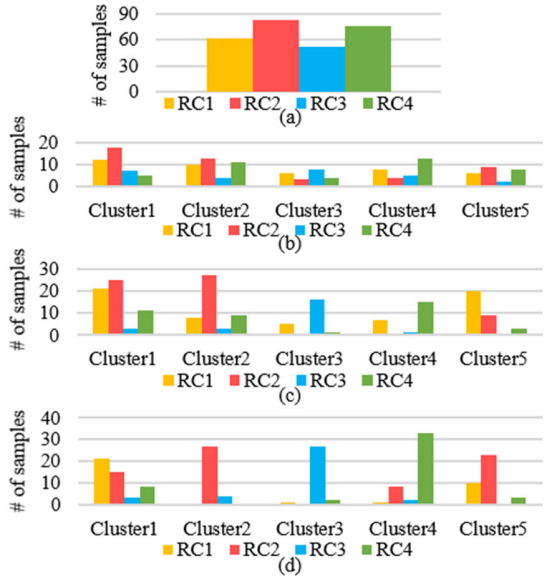

Fig. 8. Root-cause analysis results for $\mathbf{P}_{Target,1}$: (a) the histogram of four root causes labeled by human experts, and (b)-(d) the root causes associated with the five largest clusters identified by hierarchical clustering [2], two-stage clustering [7], and the proposed method respectively.

We compare the proposed method with a conventional unsupervised method based on hierarchical clustering [2] and the state-of-the-art two-stage clustering method in [7]. We quantitatively assess the performance of all methods in terms of the normalized mutual information I_{NMI} between the clustering results and the actual root causes [19], similar to the setup in [7].

A. Network Product #1

In this example, 19 network system tests are applied to both the target product $\mathbf{P}_{Target,1}$ and the source product $\mathbf{P}_{Source,1}$. 433 samples are collected for $\mathbf{P}_{Target,1}$ and 1691 samples are collected for $\mathbf{P}_{Source,1}$. Human experts have labeled all defective samples of $\mathbf{P}_{Target,1}$ with four root causes; they are used here only for the purpose of validation. No root-cause information is provided when we perform root-cause analysis using all three methods. Table I compares the proposed method with other conventional approaches based on I_{NMI}. As shown in Table I, the proposed method achieves the highest value of I_{NMI}, demonstrating its superior performance among all three methods.

Fig. 8 further shows the root-cause-analysis results for all three methods. The histogram of four root causes, denoted as RC1 to RC4, is shown in Fig. 8(a) for the defective samples of $\mathbf{P}_{Target,1}$. Fig. 8(b)-(d) show the root-cause distributions associated with the five largest clusters identified by the three different methods. In Fig. 8(b), the four root causes are not clearly distinguished in each cluster. On the other hand, each cluster in Fig. 8(c)-(d) contain only 1~2 dominant root causes, thereby demonstrating the higher accuracy of the two-stage clustering methods.

Compared to the results in Fig. 8(c), the clusters in Fig. 8(d) have higher purity. Taking Cluster3 as an example, the percentage of samples with the dominant root cause RC3 (i.e., blue bar) is higher in Fig. 8(d) than in Fig. 8(c). It demonstrates the effectiveness of our transfer-learning method in Algorithm 2.

B. Network Product #2

In this example, 17 network system tests are applied to both the target product $\mathbf{P}_{Target,2}$ with 871 samples and the source product $\mathbf{P}_{Source,2}$ with 4188 samples. Human experts have labeled all defective samples of $\mathbf{P}_{Target,2}$ with four possible root causes as the ground truth. Similar to the previous example, the proposed method outperforms the other two methods with the highest value of I_{NMI}, as shown in Table II.

More detailed results of root-cause analysis are shown in Fig. 9 for all the methods. Four root causes, denoted as RC1 to RC4, are labeled by human experts for the defective samples of $\mathbf{P}_{Target,2}$ and the corresponding histogram is shown in Fig. 9(a). Fig. 9(b)-(d) show the histograms of root causes associated with the five largest clusters diagnosed by the three different methods. In Fig. 9(b), the root causes are not clearly distinguished in each cluster. In contrast, each cluster in Fig. 9(c)-(d) contains only 1~2 dominant root causes. Moreover, the clusters in Fig. 9(d) have higher purity. Taking Cluster2 as an example, the percentage of samples with the dominant root cause RC2 (i.e., red bar) is higher in Fig. 9(d) than in Fig. 9 (c), which demonstrates the efficacy of transfer learning in Algorithm 2.

TABLE II
PERFORMANCE OF ROOT-CAUSE ANALYSIS ON NETWORK PRODUCT #2

Method	I_{NMI}	K	Runtime (s)
Hierarchical clustering [2]	0.086	40	10.8
Two-stage clustering [7]	0.161	15	13.1
Proposed method	0.250	9	86.9

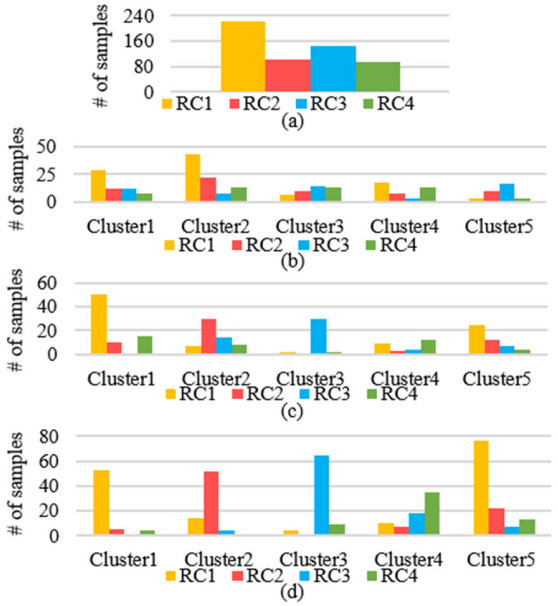

Fig. 9. Root-cause analysis results for $\mathbf{P}_{Target,2}$: (a) the histogram of four root causes labeled by human experts, and (b)-(d) the root causes associated with the five largest clusters identified by hierarchical clustering [2], two-stage clustering [7], and the proposed method respectively.

VI. CONCLUSIONS

We have proposed an unsupervised root-cause-analysis method with transfer learning for integrated systems. The proposed method is composed of two novel ideas. First, a two-stage clustering approach has been proposed by leveraging model selection based on the Silhouette score. Second, a transfer-learning method based on ensemble learning has been adopted to select valuable data from a source product to improve the root-cause-analysis accuracy on the target product. As demonstrated by two case studies from industry designs and commercial products, the proposed method has outperformed other state-of-the-art methods. In our future work, we will further consider categorical test items and enhance the robustness for transfer learning.

REFERENCES

[1] M. Palattella *et al.*, "Internet of things in the 5G era: enablers, architecture, and business models," *IEEE J-SAC*, vol. 34, no. 3, pp. 510-527, 2016.

[2] A. Gómez-Andrades *et al.*, "Automatic root cause analysis for LTE networks based on unsupervised techniques," *IEEE Trans. on VT*, vol. 65, no. 4, pp. 2369-2386, 2015.

[3] F. Ye *et al.*, "Board-level functional fault diagnosis using artificial neural networks, support-vector machines, and weighted-majority voting," *IEEE Trans. on CAD*, vol. 32, no. 5, pp. 723-736, 2013.

[4] F. Ye *et al.*, "Adaptive board-level functional fault diagnosis using incremental decision trees," *IEEE Trans. on CAD*, vol. 35, no. 2, pp. 323-336, 2015.

[5] M. Liu *et al.*, "Board-level functional fault identification using streaming data," *VTS*, pp.1-6, 2019.

[6] Y. Lei *et al.*, "An intelligent fault diagnosis method using unsupervised feature learning towards mechanical big data," *IEEE Trans. on IE*, vol. 63, no. 5, pp. 3137-3147, 2016.

[7] R. Pan *et al.*, "Unsupervised root-cause analysis for integrated systems," *ITC*, pp. 1-10, 2020.

[8] J. Pan and Q. Yang, "A survey on transfer learning," *IEEE Trans. on KDE*, vol. 22, no. 10, pp. 1345-1359, 2009.

[9] F. Ye *et al.*, "Knowledge discovery and knowledge transfer in board-level functional fault diagnosis," *ITC*, pp. 1-10, 2014.

[10] M. Liu *et al.*, "Knowledge transfer in board-level functional fault identification using domain adaptation," *ITC*, pp. 1-10, 2019.

[11] B. Liu *et al.*, "Clustering through decision tree construction," *CIKM*, pp. 20-29, 2000.

[12] C. Bishop, *Pattern Recognition and Machine Learning*, Springer, 2006.

[13] L. Kaufman and P. Rousseeuw, *Finding Groups in Data: An Introduction to Cluster Analysis*, John Wiley & Sons, 2009.

[14] T. Kamishima *et al.*, "TrBagg: a simple transfer learning method and its application to personalization in collaborative tagging," *ICDM*, pp. 219-228, 2009.

[15] A. Ramanathan *et al.*, "Integrating symbolic and statistical methods for testing intelligent systems: applications to machine learning and computer vision," *DATE*, pp. 786-791, 2016.

[16] Y. You *et al.*, "Diagnosis technique for Clustered Multiple Transition Delay Faults", *ITC-Asia*, pp. 53-58, 2020.

[17] A. Gaita *et al.*, "A SIFT-based Waveform Clustering Method for aiding analog/mixed-signal IC Verification", *ETS*, pp. 1-2, 2020.

[18] J. Han *et al.*, "Mining frequent patterns without candidate generation: a frequent-pattern tree approach," *Data Mining and Knowledge Discovery*, vol. 8, no. 1, pp. 53-87, 2004.

[19] C. Manning *et al.*, *Introduction to Information Retrieval*, Cambridge University Press, 2009.

[20] S. Salvador and P. Chan, "Determining the number of clusters/segments in hierarchical clustering/segmentation algorithms," *ICTAI*, pp. 576-584, 2004.

[21] P. Tan *et al.*, *Introduction to Data Mining*, Pearson Education India, 2016.

[22] K. Weiss *et al.*, "A survey of transfer learning," *Journal of Big Data*, vol. 3, no. 9, pp. 1-40, 2016.