Efficient Board-Level Functional-Fault Diagnosis with Missing Syndromes*

Shi Jin†, Fangming Ye†, Zhaobo Zhang‡, Krishnendu Chakrabarty†, and Xinli Gu‡
†ECE Dept., Duke University, Durham, NC, USA ‡Huawei Technologies Co. Ltd., San Jose, CA, USA
{sj137,fy11, krish}@ee.duke.edu {Zhaobo.sc.Zhang, Xinli.Gu}@huawei.com

Abstract— **Functional fault diagnosis is widely used in board manufacturing to ensure product quality and improve product yield. Advanced machine-learning techniques have recently been advocated for reasoning-based diagnosis; these techniques are based on the historical record of successfully repaired boards. However, traditional diagnosis systems fail to provide appropriate repair suggestions when the diagnostic logs are fragmented and some error outcomes, or syndromes, are not available during diagnosis. We describe the design of a diagnosis system that can handle missing syndromes and can be applied to four widely used machine-learning techniques. Several imputation methods are discussed and compared in terms of their effectiveness for addressing missing syndromes. Moreover, a syndrome-selection technique based on the minimum-redundancy-maximum-relevance ($mRMR$) criteria is also incorporated to further improve the efficiency of the proposed methods. Two large-scale synthetic data sets generated from the log information of complex industrial boards in volume production are used to validate the proposed diagnosis system in terms of diagnosis accuracy and training time.**

I. Introduction

The prevalence of high-density, complex, and high-frequency designs in today's printed-circuit boards is resulting in defects that are subtle and difficult to diagnosis using traditional techniques [**?**], [1], [2]. When faulty boards are returned to the service center for repair, debug technicians attempt to detect and diagnose failures early in the test process (e.g., through component and structural tests). A defect's most probable cause, logic location, and even physical location can sometimes be determined based on scan test [3]. However, traditional structural test techniques are not sufficient to guarantee product quality. Some defects are not detected until functional tests are carried out on the board [4].

Functional test is useful for targeting defects that cannot be easily detected by structural test. A challenging and common scenario in board-level testing is that all chips on a board pass automated test equipment (ATE) tests, but the board fails during functional test or when an application is booted up [5], [6]. The reason for such "No Trouble Found" (NTF) occurrences is that the board-level test environment is considerably different from that at the chip level. At the chip vendor location, chips are tested in a standalone mode in a controlled environment, but additional issues, such as signal integrity, power-supply noise and crosstalk, must be considered during board-level test [5]. Board-level functional testing is also challenging because of limits on controllability and observability in functional mode [7].

Unlike chip-level fault diagnosis, which can be automated via ATE and advanced debug/diagnosis tools [8], traditional board-level functional diagnosis still requires a considerable amount of manual expertise. As a result, a number of advanced board-level functional-fault diagnosis methods have recently been presented in the literature, among which rule-based, model-based, and reasoning-based learning techniques have been especially popular over the past decade. Rule-based diagnosis methods take the form "If (syndromes), then (fault)" to locate root causes [9], but there exist bottlenecks in acquiring sufficient knowledge to build an adequate set of rules for realistic designs. Model-based methods rely on an approximate representation of a real system [10], but the diagnosis models become prohibitively complex and hard to develop for large systems.

Reasoning-based methods are promising because a detailed system model is not needed to construct the diagnosis system [9], [11]. The diagnosis engine is incrementally built based on a database of successfully repaired faulty boards [11]–[14]. Machine-learning techniques facilitate reasoning-based diagnosis, providing the benefits of ease of implementation, high diagnosis accuracy, and continuous learning. The repair/replacement of faulty components is suggested through a ranked list of suspect components, e.g., based on artificial neural networks (ANNs), support-vector machines (SVMs), and decision trees (DTs) [15], [16]. However, the diagnosis accuracy of these methods may be significantly reduced when the repair logs are fragmented and some errors, or syndromes, are not available during diagnosis. Since root-cause isolation for a failing board relies on reasoning based on syndromes, any information loss (e.g., missing syndromes) during the extraction of a functional diagnosis log may lead to ambiguous repair suggestions.

In this paper, we propose a board-level functional diagnosis system that applies different methods to handle missing syndromes in various machine learning models. The syndromes from a faulty-board's functional diagnosis log are analyzed and *preprocessed* before root-cause isolation. A feature selection process is also incorporated into our preprocessing engine to reduce redundant, irrelevant, or even misleading syndromes introduced by traditional missing-syndrome-handling methods. We compare these methods in terms of their effectiveness in handling missing syndromes.

The remainder of this paper is organized as follows. Section II discusses the missing-syndrome problem in more detail. Section III discusses the effects of missing syndromes on a functional diagnosis system. A number of methods are used to empower functional diagnosis system to handle missing

*This research was supported by a grant from Huawei Technologies Co. Ltd. A preliminary version of this paper was published in Proc. IEEE Asian Test Symposium, 2013.

syndromes. These techniques have been implemented using the WEKA framework [17]. In Section IV, experimental results on two synthetic boards and two industrial boards in volume production, are used to demonstrate the effectiveness of the proposed missing-syndrome-tolerant functional diagnosis system in terms of diagnosis accuracy and training time. Finally, Section V concludes the paper.

II. Problem Statement and Paper Contributions

A functional diagnostic system based on machine learning does not need to understand the complex functionality of boards, and it is able to automatically derive and exploit knowledge from the repair logs of previously documented cases. A flowchart of such a diagnosis system is shown in Fig. 1. The extraction of fault syndromes, i.e., test outcomes, is critical for model training in a reasoning-based diagnosis system. The fault syndromes should provide a complete description of the failure, and the extracted syndromes for different actions should have sufficient diversity such that we can eliminate ambiguity in the eventual repair recommendations. All the test-outcome information is stored in a log file. For example, a segment of the log file for a failed traffic test is shown in Fig. 2(a). The fault syndromes extracted from this log are R3d3metro (mis-matched interface), LA1 Engine 0x0000 0fff (error counter), and 0x11 (error code). Each of these elements is considered to be one syndrome. These fault syndromes record any abnormal value obtained from functional tests, including transfer value mismatches, register dumps, or loop failures. Moreover, the outcomes from analog indicators, such as power and temperature monitors, are also abstracted and logged as syndromes. The repair action is often directly recorded in the database, e.g., “replacement of component U11”. All the extracted syndromes and actions are used for training of the proposed diagnosis system.

In prior work, e.g., in [15], functional-test logs were parsed in a coarse manner where a syndrome is 1 (*fail*) if a functional test leads to an error at the corresponding observation point. All other functional tests and observation-point pairs, which are mapped to syndromes, are deemed as 0 (*pass*); see Fig. 2(b) [15], [16], [18]. However, a syndrome should not be denoted as either *fail* or *pass* when the observation of the test is not available in practice. We propose to define a third syndrome, namely *missing* syndrome, when the syndrome for a (test, observation point) pair is not observed (see Fig. 2(c)). There are two scenarios that lead to the absence of observation

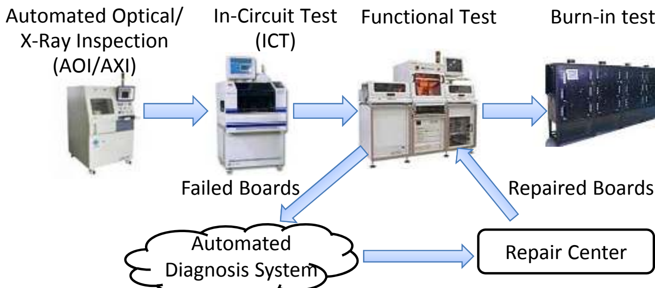


Fig. 1: Illustration of automated board diagnosis.

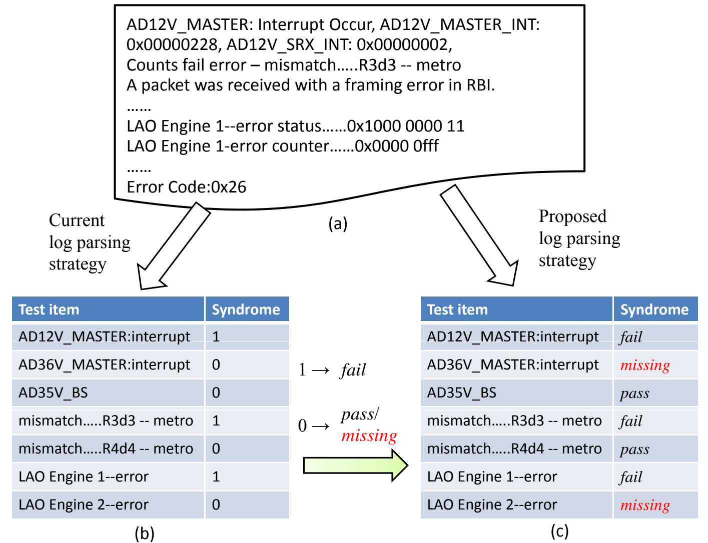


Fig. 2: Illustration of parsing a log with and without *missing* syndromes.

of a syndrome:

1) **Systematic missing syndromes**: In current test-program design methods, the tasks of designing functional tests are divided into multiple packages and dispatched to multiple design teams [4]. Functional test packages are executed sequentially on a board-under-test. There is a likelihood that the execution of functional tests in later packages is affected by tests in early packages. For example, a functional test, *mainboost*, is used to check the functionality of a global controller, which is required to enable the functional tests that follow [4]. If the controller fails or the attached DRAMs fail, the diagnosis procedure is terminated and subsequent tests cannot be executed. The syndromes corresponding to these tests are, therefore, not available to the diagnosis system.

2) **Random missing syndromes**: Another scenario in current diagnosis systems is that a few functional tests may occasionally fail to record syndromes. The reason for these occasional syndrome escapes may not be due to board failure, but because of program bugs in functional test design [4], [11]. Such bugs may either lead to incorrectly recorded syndromes or missing observations in the log. In these scenarios, the diagnosis system should be designed to be able to reason in the presence of these missing syndromes.

Both of the above scenarios lead to a decrease in the total number of syndromes available for diagnosis, thus resulting in low diagnosis accuracy. In addition, although several machine-learning techniques such as ANNs are considered to be noise-tolerant, noisy input is different from input containing missing values. Noisy input still contains useful information that can be used by machine learning models while cases containing missing values are typically deleted from the data set. Most machine learning-based diagnosis systems proposed in the literature are not equipped to deal with these missing values [18]–[20]. We therefore address this important practical problem of handling missing syndromes in a machine-learning-based functional diagnosis system in the next section.

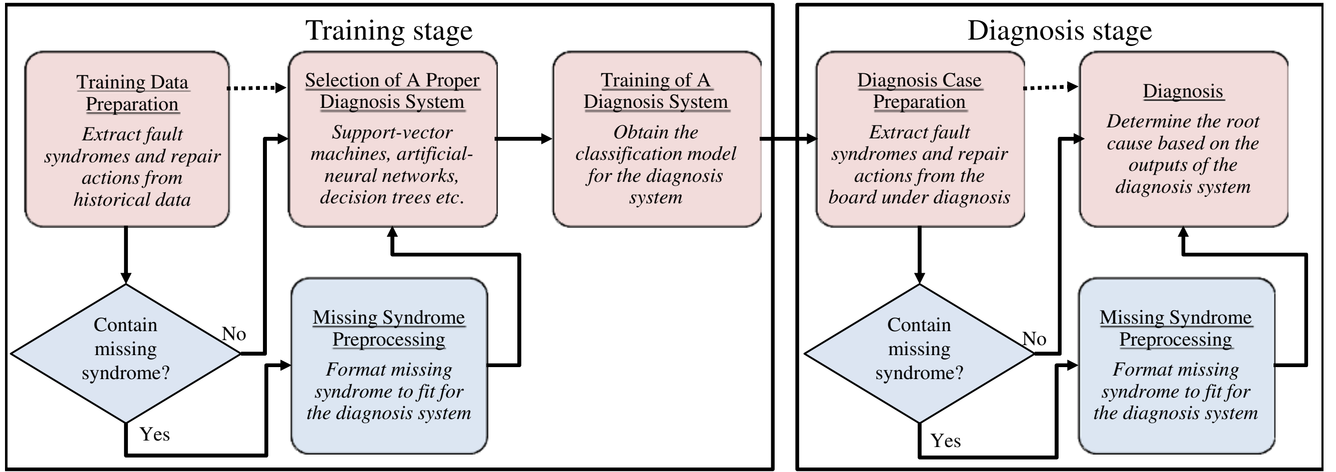


Fig. 3: The reasoning-based diagnosis flow with the capability of handling missing syndromes.

III. Methods to Handle Missing Syndromes

Missing data frequently occur in applied data analysis [21], [22]. As described in Section II, several scenarios may lead to missing syndromes, e.g., resulting in systematic missing syndromes and random missing syndromes. The effectiveness of a specific method for handling missing syndromes depends not only on its inherent properties, but also on the machine-learning models it is used with. In this section, we describe five kinds of missing handling methods as well as a general diagnosis flow to address the problem of missing syndromes in board-level fault diagnosis.

A. Missing-syndrome-tolerant fault diagnosis flow

A reasoning-based diagnosis flow in [15] can be improved by integrating the component of preprocessing missing values, as shown in Fig. 3. A revised diagnosis flow that is capable of handling missing syndromes can be described as follows. Once training data are extracted from diagnosis logs, the diagnosis system determines whether the data set contains missing syndromes, i.e., by comparing the number of obtained syndromes and total number of designed syndromes. If no missing syndrome is detected, the diagnosis system continues with the standard diagnosis-system training. Otherwise, the training data set is handled by the missing-value-preprocessing component. The processed data set is then sent for training of the diagnosis system. The same procedure also applies to the new failing boards under diagnosis. The preprocessed new cases are sent to the diagnosis system for root-cause isolation.

The missing-value-preprocessing component contains five parallel subcomponents for handling missing syndromes: complete-case analysis, fractional instances, numerical imputation, label imputation and feature selection. These subcomponents could either be chosen automatically based on the type of machine learning algorithms used in current diagnosis system, or be set manually by users themselves. Moreover, these subcomponents could even be deliberately combined to better match the characteristics of the current system, leading to higher diagnosis accuracy.

B. Missing-syndrome-preprocessing methods

1) Complete-Case Analysis: The basic idea of complete-case analysis is to ignore all the missing values and base the analysis only on complete-case data. Since this method is widely used in the Naive Bayes (NB) classifier, we use a simple NB-based fault diagnosis example to show how this method address missing syndromes.

First, suppose we have a set of faulty boards with two candidate root causes A_1 and A_2, and we encode them as $y = -1$ and $y = 1$, respectively. Here, we merge the syndromes and the known root causes into a matrix $\mathcal{M} = [\mathcal{B}|\mathcal{C}]$, where the left ($\mathcal{B}$) side refers to syndromes, while the right side ($\mathcal{C}$) refers to the corresponding fault classes. This matrix represents the training information for a diagnosis system that does not contain any missing syndromes,

$$\mathcal{M} = \left[\begin{array}{cccc:c} 1 & 1 & 0 & 1 & 1 \\ 0 & 1 & 1 & 0 & -1 \\ 0 & 0 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 & -1 \\ 1 & 0 & 0 & 0 & -1 \\ 1 & 1 & 1 & 0 & 1 \end{array}\right], \tag{1}$$

A Naive Bayes (NB) classifier is a Bayes' theorem-based probabilistic model, which makes inferences from historical data and a *prior* distribution [23]. The final diagnosis step of a NB classifier is to pick the root-cause candidate with maximal *posterior* occurrence probability.

Assuming that a new faulty board with syndrome set $T = \{1, 1, 1, 1\}$ is fed into our NB-based diagnosis system, the *posterior* occurrence probability of this new case can be calculated as shown below:

$$p(A_1|T)=\frac{p(T|A_1) \times p(A_1)}{p(T)} = 0.800$$

$$p(A_2|T)=\frac{p(T|A_2) \times p(A_2)}{p(T)} = 0.200$$

Thus, A_1 has higher *posterior* occurrence probability and would be chosen as the root cause for the new faulty board.

Next, consider the training matrix $\mathcal{M}$ containing missing syndromes, and let ? denotes the missing values:

$$\mathcal{M} = \left[\begin{array}{cccc:c} 1 & 1 & 0 & ? & 1 \\ ? & 1 & 1 & 0 & -1 \\ 0 & ? & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 & -1 \\ 1 & 0 & ? & 0 & -1 \\ 1 & 1 & 1 & 0 & 1 \end{array}\right] \quad (2)$$

The way that the complete-case analysis addresses these missing syndromes is to remove them from training data during the calculation of $posterior$ occurrence probability. Using the same test case $T = \{1, 1, 1, 1\}$, the $posterior$ occurrence probability of this case is calculated as below:

$$p(\mathrm{A}_1|\mathrm{T}) = \frac{p(\mathrm{T}|\mathrm{A}_1) \times p(\mathrm{A}_1)}{p(\mathrm{T})} = 0.667$$

$$p(\mathrm{A}_2|\mathrm{T}) = \frac{p(\mathrm{T}|\mathrm{A}_2) \times p(\mathrm{A}_2)}{p(\mathrm{T})} = 0.333$$

The above result shows that A_1 is still the most likely root cause for the test case. Thus, in the above example, when the training data contains several missing syndromes, after applying complete-case analysis, our NB classifier can still give accurate fault diagnosis.

Complete-case analysis can also be used with other machine learning models such as SVM and ANN. However, complete-case analysis in these models will discard the entire instances containing missing values instead of just ignoring the missing values themselves, resulting in much lower diagnosis accuracy than the NB classifier when the amount of missing information is relatively high.

2) Fractional Instances: The goal of the fractional-instances method is to distribute missing syndromes over other complete instances. An instance containing missing values will be split into multiple weighted instances. Each weighted instance represents one possible value combination of those missing syndromes. The weight value associated with each split instance is the fraction of complete cases that have the same syndrome values as this split instance.

Let us assume that we apply this fractional-instance method to the training matrix with missing syndromes shown in (2). A new matrix is obtained as shown below:

$$\mathcal{M} = \left[\begin{array}{cccc:c} 0.4 & 0.4 & 0 & 0.4 & 1 \\ 0.6 & 0.6 & 0 & 0 & 1 \\ 0.6 & 0.6 & 0.6 & 0 & -1 \\ 0 & 0.4 & 0.4 & 0 & -1 \\ 0 & 0.8 & 0.8 & 0.8 & 1 \\ 0 & 0 & 0.2 & 0.2 & 1 \\ 0 & 1 & 1 & 1 & -1 \\ 0.8 & 0 & 0.8 & 0 & -1 \\ 0.2 & 0 & 0 & 0 & -1 \\ 1 & 1 & 1 & 0 & 1 \end{array}\right]$$

The advantage of the fractional-instances method is that it greatly enriches the incomplete training data by artificially adding instances to represent every possible value combinations of missing syndromes.

However, the number of additional instances will increase exponentially with the number of missing values, making this method infeasible for machine-learning algorithms that require training data using the entire syndrome set, e.g., traditional ANNs and SVMs. In contrast, the decision tree (DT), which is a tree-like predictive model for mapping observations of an item to its targeted value [24], is suitable for this method since a DT typically only needs a subset of syndromes for training.

3) Numerical Imputation: Numerical imputation is widely used to infer missing values. A *missing* value is predicted on a numerical scale based on analysis of the failing boards. Several imputation methods, e.g., mean and mode, can be used [21], [25] and they are described in detail later. Suppose that we have the same training matrix with missing syndromes as shown in (2). After applying numerical imputation, the new matrix is given by (3)

$$\mathcal{M} = \left[\begin{array}{cccc:c} 1 & 1 & 0 & b_{41} & 1 \\ b_{12} & 1 & 1 & 0 & -1 \\ 0 & b_{23} & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 & -1 \\ 1 & 0 & b_{35} & 0 & -1 \\ 1 & 1 & 1 & 0 & 1 \end{array}\right] \quad (3)$$

where b_{ij} is the missing value of syndrome t_i collected from the failing board j. Different imputation methods can be used to assign values for b_{ij}; these are discussed below.

I) Zero Imputation: In the prior log parsing strategy in Fig. 2(b), all *missing* syndromes are deemed to be *pass*, which is denoted as 0. Zero imputation follows the same imputation method, where any missing syndrome b_{ij} is imputed with 0. As an illustration, (3) can be revised with zero-imputation values, as shown below:

$$\mathcal{M} = \left[\begin{array}{cccc:c} 1 & 1 & 0 & \mathit{0} & 1 \\ \mathit{0} & 1 & 1 & 0 & -1 \\ 0 & \mathit{0} & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 & -1 \\ 1 & 0 & \mathit{0} & 0 & -1 \\ 1 & 1 & 1 & 0 & 1 \end{array}\right], \quad (4)$$

where italics indicate the imputed values and are used to denote imputed values throughout this paper.
This type of a missing-syndrome-handling method is widely used in the Artificial Neural Network (ANN) model, which is a supervised machine-learning method that has also been widely used for pattern classification and related problems [26], [27].

II) Mode-Value Imputation: Assume that A_i is the full set of entries for syndrome x_i, A'_i is the subset of A_i that contains all the non-missing entries, where $a_{ij} \neq missing$, and $a_{ij} \in A_i$. In the mode-value imputation method, we use the following equation to impute missing values.

$$b_{ij} = mode(A'_i) = \begin{cases} 1, & \forall\ P(a_{ij} = 1, a_{ij} \in A'_i) > 0.5 \\ 0, & \forall\ P(a_{ij} = 1, a_{ij} \in A'_i) \leq 0.5, \end{cases}$$

where the imputation value b_{ij} is the same for each syndrome. As an illustration, the corresponding synthetic

data in (3) is revised as:

$$\mathcal{M} = \left[\begin{array}{cccc:c} 1 & 1 & 0 & \mathit{0} & 1 \\ \mathit{1} & 1 & 1 & 0 & -1 \\ 0 & \mathit{1} & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 & -1 \\ 1 & 0 & \mathit{1} & 0 & -1 \\ 1 & 1 & 1 & 0 & 1 \end{array}\right] \quad (5)$$

Mode imputation may be ineffective when the probability of the fault syndrome being '0' or '1' are similar. This method will always impute the value with higher occurrence probability, which may distort the original distribution.

III) Mean-Value Imputation: Assume that A_i is the complete set of entries for syndrome x_i, and A'_i is the subset of A_i that contains all the non-missing entries, where $a_{ij} \neq missing$ and $a_{ij} \in A_i$. In this method, a mean value b_{ij} is computed based on A'_i, such that the imputation value b_{ij} can be regarded as the probability that the syndrome x_i is *fail*. The imputation value b_{ij} is calculated as shown below:

$$b_{ij} = mean(A'_i) = \frac{\sum_{a_{ij} \in A'_i} a_{ij}}{|A'_i|}, \quad (6)$$

where the imputation value b_{ij} is same for each syndrome x_i. As an illustration, the corresponding matrix for the synthetic data is updated as:

$$\mathcal{M} = \left[\begin{array}{cccc:c} 1 & 1 & 0 & \mathit{0}.4 & 1 \\ \mathit{0}.6 & 1 & 1 & 0 & -1 \\ 0 & \mathit{0}.8 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 & -1 \\ 1 & 0 & \mathit{0}.8 & 0 & -1 \\ 1 & 1 & 1 & 0 & 1 \end{array}\right] \quad (7)$$

The combination of mean and mode imputation is widely used in the Support-Vector Machine (SVM) model, which is a supervised machine-learning technique [28] and its goal is to define an optimal separating hyperplane (OSH) to separate two classes. Generalizations also exist for handling more than two classes. For features having continuous values, the SVM's default mechanism inserts mean values into missing positions while for features with discrete values, it imputes mode values to replace missing syndromes.

4) Label Imputation: The numerical format of syndromes shown in Fig. 2(b) is no longer sufficient to describe the observations in the log of functional tests when we introduce a third syndrome. Therefore, we propose to use another label, referred to as *missing*, to denote the absence of a syndrome; see Fig. 2(c). Equation (3) can now be updated as Equation (8).

$$\mathcal{M} = \left[\begin{array}{cccc:c} pass & pass & fail & missing & 1 \\ missing & pass & pass & fail & -1 \\ fail & missing & pass & pass & 1 \\ fail & pass & pass & pass & -1 \\ pass & fail & missing & fail & -1 \\ pass & pass & pass & fail & 1 \end{array}\right] \quad (8)$$

Machine-learning techniques can handle label syndromes by converting labels to numerical values [21]. For a ternary syndrome, two values are usually used to represent one syndrome. The first value a_{ij} still denotes the failure observed in a syndrome, such that it is 1 if a syndrome is observed as *fail*, and 0 otherwise. The second value a'_{ij} denotes the occurrence of a syndrome, such that a missing syndrome is denoted with a weighted value d, and 0 otherwise. The reason for using these weighted values is that the additional feature a'_{ij} may mask the relationship between the original syndrome-fault pair. Fault syndromes are usually the dominant features for root-cause isolation, while missing syndromes are the secondary features. Therefore, it is necessary to select an appropriate value d in label imputation. The updated synthetic board is as follows:

$$\mathcal{M} = \left[\begin{array}{cccccccc:c} 1 & (0) & 1 & (0) & 0 & (0) & 0 & (d_{41}) & 1 \\ 0 & (d_{12}) & 1 & (0) & 1 & (0) & 0 & (0) & -1 \\ 0 & (0) & 0 & (d_{23}) & 1 & (0) & 1 & (0) & 1 \\ 0 & (0) & 1 & (0) & 1 & (0) & 1 & (0) & -1 \\ 1 & (0) & 0 & (0) & 0 & (d_{35}) & 0 & (0) & -1 \\ 1 & (0) & 1 & (0) & 1 & (0) & 0 & (0) & 1 \end{array}\right] \quad (9)$$

where d_{ij} is the weighted value for the jth fault syndrome of the ith instance. We use the following weighted mean-value imputation to calculate d_{ij},

$$d_{ij} = \frac{mean(A'_j)}{w} = \frac{\sum_{a_{ij} \in A'_i} a_{ij}}{w|A'_i|}, \quad (10)$$

where w is a single weight for tuning the feature of missing syndromes, and it is the same for imputing all syndromes. Choosing an appropriate value of w is important for a diagnosis system in order to achieve high diagnosis accuracy. We provide guidelines for selecting w in Section IV.

C. Feature Selection

Feature selection, also referred as subset selection, is used to select an effective, but reduced, set of syndromes for use by a diagnosis system. The main idea of methods for handling missing syndromes is to add statistical information to the incomplete data set to compensate for losses caused by missing syndromes. However, the additional information may also introduce irrelevant, redundant, or even misleading syndromes, lowering the diagnosis accuracy of the original system.

Therefore, the goal of feature selection is to identify a set of most important features from incomplete data for characterization. One of the most popular solutions for the subset-selection problem is based on the metric of minimum-redundancy-maximum-relevance ($mRMR$). Suppose we have a set of successfully repaired faulty boards with root cause set $\mathbf{A} = \{A_1, A_2, ..., A_N\}$ and syndrome set $\mathbf{T} = \{T_1, T_2, ..., T_M\}$. For each target root cause A_i and a given syndrome T_j, their

mutual information $E(A_i, T_j)$ is calculated as shown in (11):

$$E(A_i, T_j) = -p(A_i|t_j) \log p(A_i|t_j) - p(A_i|\overline{t_j}) \log p(A_i|\overline{t_j}) \tag{11}$$

where t_j is the event that syndrome T_j manifests itself, and $\overline{t_j}$ is the complementary event. We then calculate the relevance value $D(\mathbf{A}, T_j)$ between the syndrome T_j and root-cause set $\mathbf{A}$ as follows:

$$D(\mathbf{A}, T_j) = \frac{1}{|\mathbf{A}|} \sum_{A_i \in \mathbf{A}} E(A_i, T_j) \tag{12}$$

The MaxRel set $\mathbf{T'} = \{T'_1, T'_2, ..., T'_m\}$ is a selected subset of the top m syndromes having highest relevance value. The set $\mathbf{T'}$ is further evaluated by computing its redundancy value $R(\mathbf{T'})$ as shown below:

$$R(\mathbf{T'}) = \frac{1}{|\mathbf{T'}|^2} \sum_{T'_i \in \mathbf{T'}} \sum_{T'_j \in \mathbf{T'}} E(T'_i, T'_j) \tag{13}$$

where $E(T'_i, T'_j), T'_i \neq T'_j$, is the mutual information between T'_i and T'_j. We then calculate the minimum-redundancy maximum-relevance ($mRMR$) value as follows:

$$mRMR(\mathbf{T'}) = \frac{1}{|\mathbf{T'}|^2} \sum_{T'_k \in \mathbf{T'}} D(\mathbf{A}, T'_k) - R(\mathbf{T'}). \tag{14}$$

We can next determine the minimum-redundancy-maximum-relevance ($mRMR$) syndrome subset $\mathbf{T*}$ with the largest $mRMR$ value, as follows:

$$\mathbf{T*} = \max_{\mathbf{T'}} \{mRMR(\mathbf{T'})\}. \tag{15}$$

We can now discuss techniques for handling missing syndromes in the $mRMR$ subset selection process. Two approaches are discussed below:

1) Complete-Case Analysis: The first method that we use to address missing syndromes in $mRMR$ subset selection is the complete-case analysis as described in Section III-B1. Suppose we have a set of successfully repaired faulty boards with root cause set $\mathbf{A} = \{A_1, A_2, ..., A_N\}$ and syndrome set $\mathbf{T} = \{T_1, T_2, ..., T_M\}$. We can compute the desired *posterior* occurrence probability of root cause A_j, $p(A_j|T)$, using Equation (**??**). Then, the feature selection approach can calculate the relevance values, redundancy values, and $mRMR$ values, using this set of *posterior* and Equations (11)-(14). The final $mRMR$ syndrome subset can then be determined by selecting syndromes with largest $mRMR$ values.

2) Label imputation: The second approach we use here is to introduce label syndromes such that each missing syndrome can be treated as a separate value. A simple example can help better illustrate how this method works. In Equation (16), the $\rightarrow$ indicates that after feature selection, the original matrix on the left side has been reduced to the matrix on the right side. We can see that only syndrome T_1 is preserved in the new syndrome set. Now, assume we have several missing values in the original matrix and we would like to treat them as separate values during feature selection. The new process is shown in Equation (17). We can see that label imputation doubles the size of the syndrome set, providing more information for the subsequent subset-selection step. In the final

TABLE I: Information about the synthetic boards and industrial boards used for classification.

	Board 1	Board 2	Board 3	Board 4
Number of syndromes	207	420	898	3974
Number of root causes	14	10	222	352
Number of boards	1400	1000	2965	5000

reduced-syndrome set, both T_1 and T_2 are preserved for future diagnosis while T_3 and other irrelevant label syndromes are removed after feature selection.

$$\mathcal{M} = \left[\begin{array}{ccc:c} 1 & 0 & 1 & 1 \\ 0 & 1 & 0 & -1 \\ 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & -1 \end{array}\right] \rightarrow \left[\begin{array}{c:c} 1 & 1 \\ 0 & -1 \\ 1 & 1 \\ 0 & -1 \end{array}\right] \tag{16}$$

$$\mathcal{M} = \left[\begin{array}{ccc:c} 1 & ? & 1 & 1 \\ ? & 1 & ? & -1 \\ 1 & ? & ? & 1 \\ ? & 1 & 0 & -1 \end{array}\right] \rightarrow \left[\begin{array}{ccccc:cc} 1 & 0 & ? & d_{21} & 1 & 0 & 1 \\ ? & d_{12} & 1 & 0 & ? & d_{32} & -1 \\ 1 & 0 & ? & d_{23} & ? & d_{33} & 1 \\ ? & d_{14} & 1 & 0 & 0 & 0 & -1 \end{array}\right] \rightarrow \left[\begin{array}{cc:c} 1 & 0 & 1 \\ 0 & 1 & -1 \\ 1 & 0 & 1 \\ 0 & 1 & -1 \end{array}\right] \tag{17}$$

The comparison between these two methods in terms of diagnosis accuracy and subset size is described in Section IV.

IV. Experiments and Results

Experiments were performed on two synthetic boards and two industrial boards. The boards under study include line process units (LPUs) in a high-end router with 10-Gbit/s interfaces, which are designed for core and backbone commercial networks. In each board, there are multiple complex chips, including network processor, algorithmic search engine, packet forwarding IC, multi-core CPU, etc. Each core chip is connected to tens of hundred-megabyte external memories, such as DDR-RAM, QDR-RAM and CAM. Other devices such as tens of PLLs, voltage regulators, voltage and temperature sensors, and hundreds of other passive components are soldered on the boards as well.

As shown in Table I, for Board 1, a total of 207 fault syndromes and 14 faulty components are extracted from failure logs. For Board 2, 420 syndromes and 10 faulty components are identified. Board 3 has 898 syndromes and 222 potential root causes. For Board 4, 3974 fault syndromes and 352 potential root causes are extracted.

The synthetic data sets are generated from two actual board designs because only a small number of faulty boards are available in the current early production phase. For each root cause, we apply regression to every syndrome, thereby generating a probability based on the occurrence of the syndrome (i.e., the syndrome with more 1's in a class will be assigned a higher probability, while a lower probability indicates that it has more 0's). Based on these probabilities, we insert the corresponding variance values to the synthetic data (i.e., the syndrome with more 1's in the original data will also have more 1's in the synthetic data). We generated 100 new cases for each repair candidate, thus the synthetic data sets consists of 1400 cases for Board 1 and 1000 cases for Board 2.

To evaluate the performance of the proposed methods for handling missing syndromes, we need to generate missing

syndromes in our training data. The occurrence of missing syndromes in current diagnosis systems is most likely due to program bugs in the interface of gathering data from test equipment instead of board failures or test-design incompleteness. Such bugs may either lead to incorrectly recorded syndromes or missing observations in the log files. Since missing syndromes of all types (and anywhere in the system response) can result due to these data-gathering-interface bugs, we assume that missing syndromes can occur in any position in our training data. Let T be the entire set of training set, S be the total number of syndromes, and C be the total number of failing boards. Let X be the total number of missing syndromes in T. The missing ratio (MR) is defined as $\frac{X}{S \times C}$. In this work, we use four values for the missing ratio, 10%, 30%, 50%, and 70%, to consider a wide range of scenarios. Since most machine-learning methods employ heuristics and the missing syndromes can be randomly distributed, Monte-Carlo simulation is necessary to ensure that our results are not favorable as a result of serendipity. We carried out 500 Monte- Carlo simulation runs because there was no significant change in the results when we increased this number to over 500. For less than 500 Monte Carlo runs, we observed significant variation in the results. The average success ratios are presented to show the diagnosis performance with different missing-syndrome-handling methods.

All the algorithms are implemented in an open-source Machine learning toolkit, namely WEKA [29]. Experiments were run on a 64-bit Linux system with 12 GB of RAM and quad-core Intel i7 processors running at 2.67 GHz.

To evaluate the diagnosis performance of different machine-learning systems, we use a 5-fold *cross-validation* method [30], which randomly partitions the training set into 5 groups. Each group is regarded as a test case while all of the other cases are fed for training. The success ratio (SR), referred to as a percentage, is the ratio of the number of correctly diagnosed cases to the total number of cases in the testing set. For example, if 10 instances are tested, SR of 70% means that 7 out of 10 cases are correctly classified. In addition, since SR is a coarse metric for multi-class classifiers (e.g., a board-level functional fault diagnosis system) and does not provide any suggestion for improving diagnosis, we introduce a fine-grained set of information-theoretic metrics, called $precision$ and $recall$, to comprehensively evaluate the diagnosis system. Positive predictive value (PPV), also known as $precision$, is the proportion of the predicted positive cases that are correct, calculated using 18:

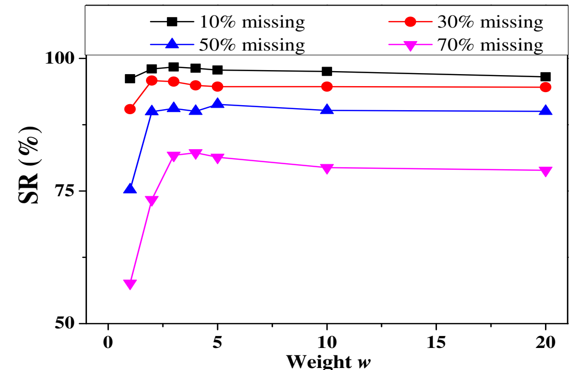
$$PPV(precision) = \frac{TP}{TP + FP} \tag{18}$$

True positive rate (TPR), also known as $recall$, is the proportion of positive cases that are correctly identified, calculated as follows:

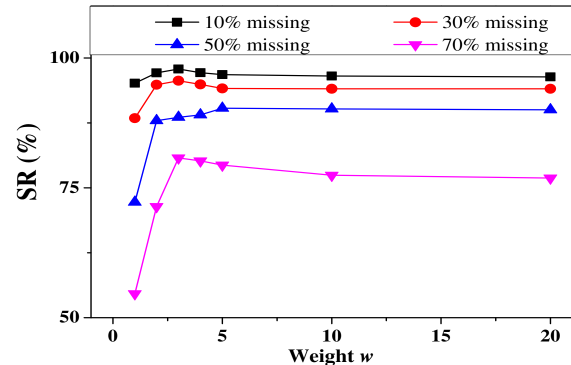
$$TPR(recall) = \frac{TP}{TP + FN} \tag{19}$$

where TP is the number of correctly predicted positive cases; FP is the number of incorrectly predicted positive cases; FN is the number of incorrectly predicted negative cases; And TN is the number of correctly predicted negative cases. In board-level diagnosis, $precision$ describes the percentage of success in predicting a root cause, while $recall$ reflects the percentage of success for a root cause to be predicted. A combination of these two metrics provides a more complete picture.

First, we evaluate the label imputation method. Different weight values, $1 \leq w \leq 20$, in label imputation are compared in terms of success ratios in diagnosis, as shown in Fig. 4-11. We observe that although the trends of weight values depend on type of machine learning models as well as missing ratios, too small or large a choice of w typically leads to reduced diagnosis accuracy. For example, for the SVM model in Board 1, if we consider 50% missing syndromes, the SR is only 75% for $w = 1$. As we increase w, the SR increases to 91% for $w = 5$. However, as we continue to increase w, the SR slightly decreases. The SR obtained by using $w = 20$ is 88%. Similar SR results are obtain for other missing ratios and learning algorithms for Board 1 in Fig. 4. The reason smaller w leads to lower SR is that the use of missing syndromes with a small value of w masks the fail syndromes; the fail syndromes are, however, more informative for diagnosis. In contrast, the reason that too large a value of w leads to lower SR is that the missing syndromes do not contribute to diagnosis in these cases. If w is infinite, the imputation value d becomes 0 for all missing syndromes.

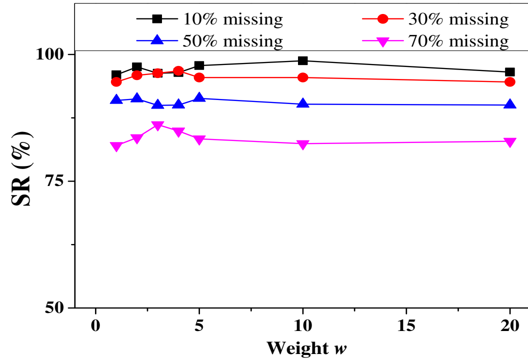


(a) SVM

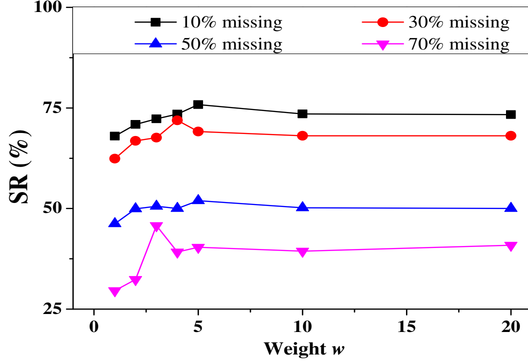


(b) ANN

Fig. 4: Comparison of SR for SVM and ANN diagnosis using label imputation with different weights for Board 1.

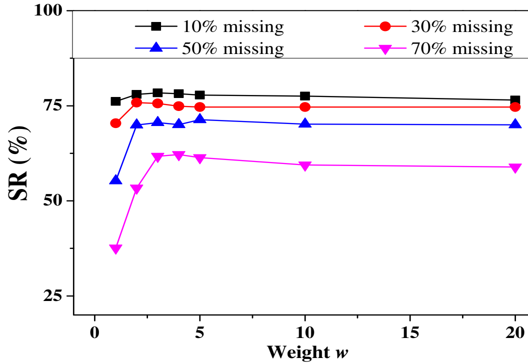


(a) Bayes

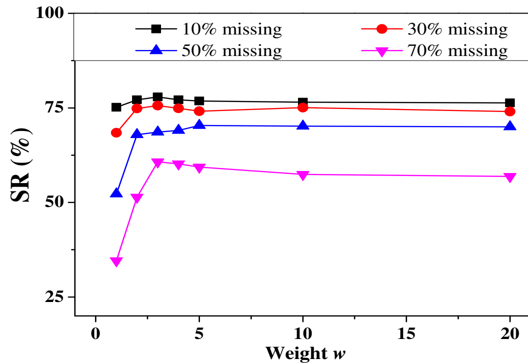


(b) DT

Fig. 5: Comparison of SR for NB and DT diagnosis using label imputation with different weights for Board 1.

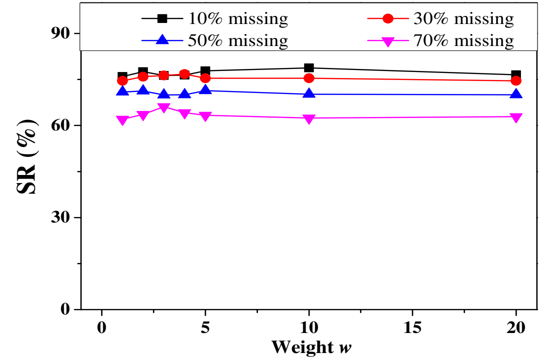


(a) SVM

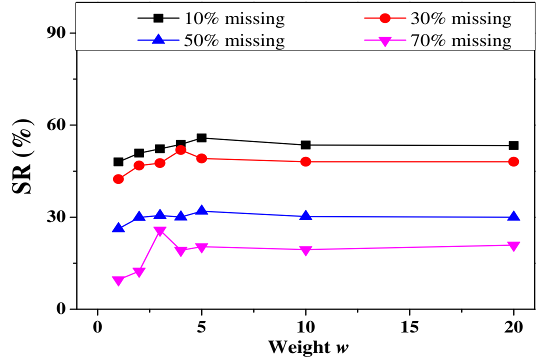


(b) ANN

Fig. 6: Comparison of SR for SVM and ANN diagnosis using label imputation with different weights for Board 2.

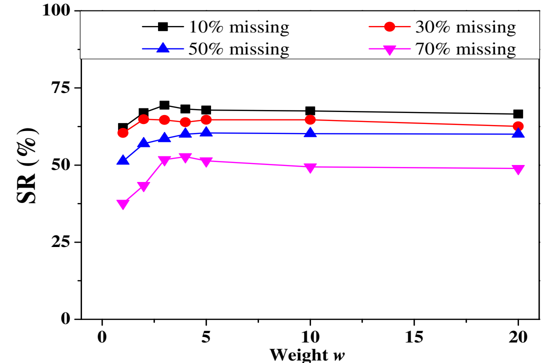


(a) Bayes

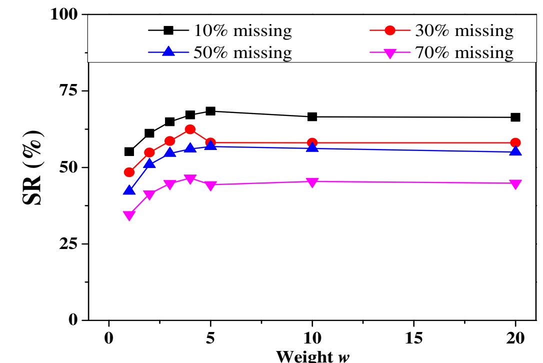


(b) DT

Fig. 7: Comparison of SR for NB and DT diagnosis using label imputation with different weights for Board 2.

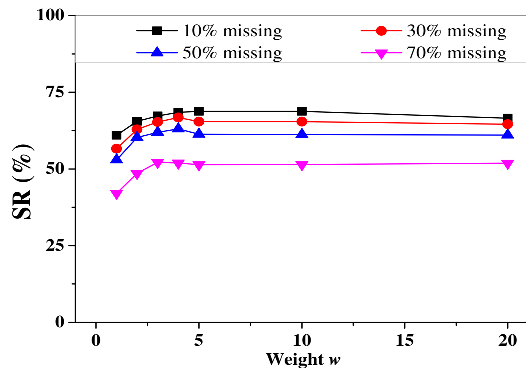


(a) SVM

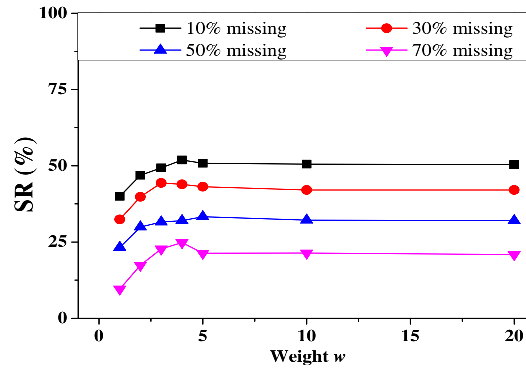


(b) ANN

Fig. 8: Comparison of SR for SVM and ANN diagnosis using label imputation with different weights for Board 3.

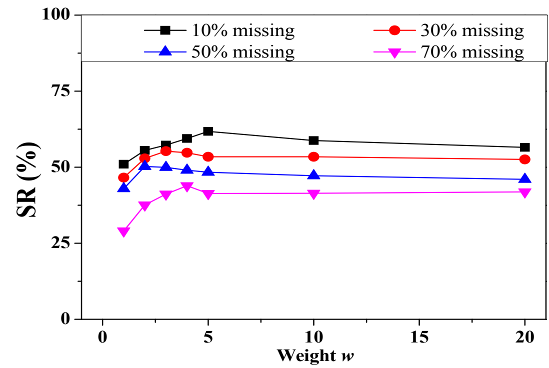


(a) Bayes

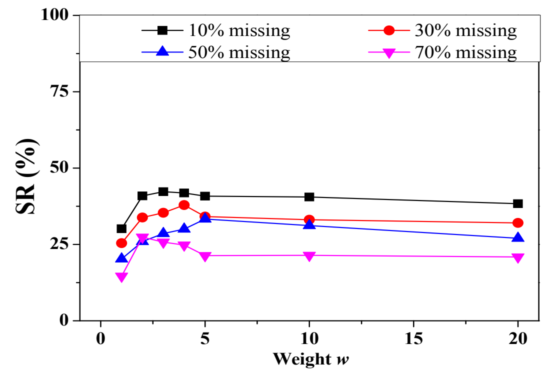


(b) DT

Fig. 9: Comparison of SR for NB and DT diagnosis using label imputation with different weights for Board 3.

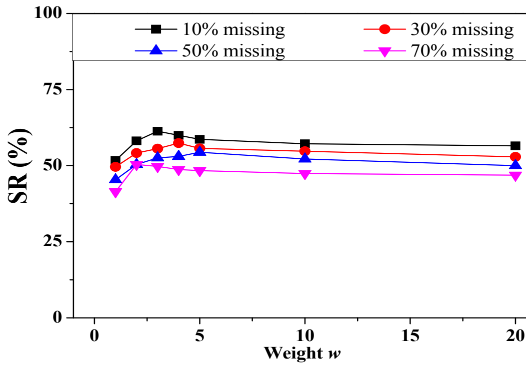


(a) Bayes

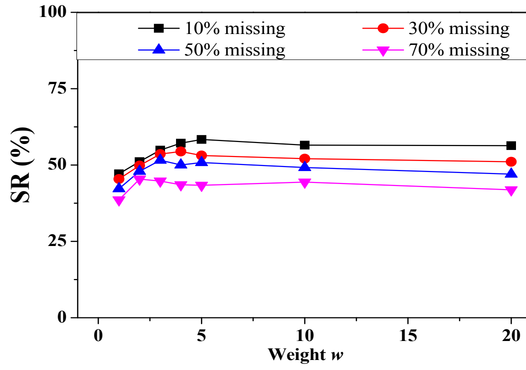


(b) DT

Fig. 11: Comparison of SR for NB and DT diagnosis using label imputation with different weights for Board 4.

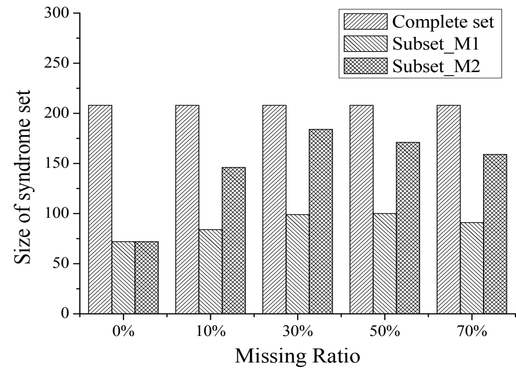


(a) SVM

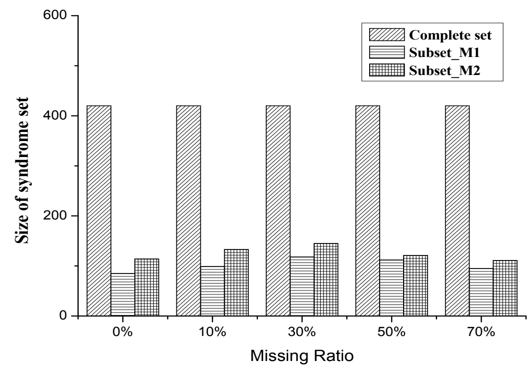


(b) ANN

Fig. 10: Comparison of SR for SVM and ANN diagnosis using label imputation with different weights for Board 4.

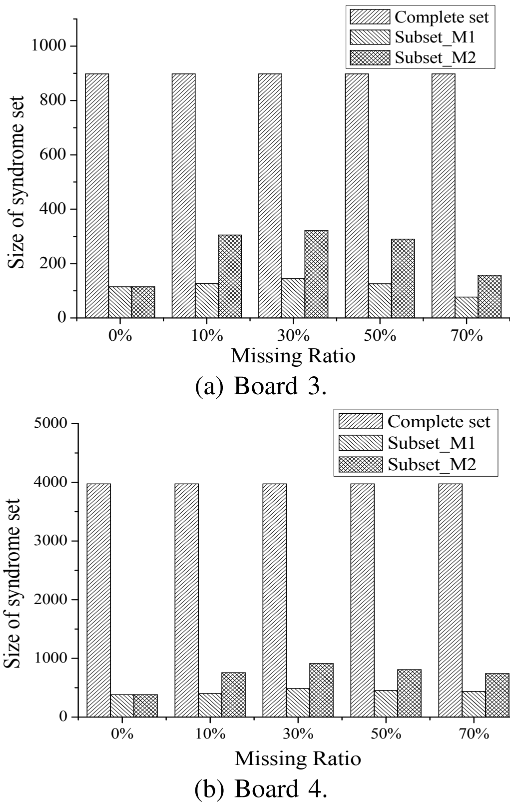


(a) Board 1.



(b) Board 2.

Fig. 12: Subset size of two feature-selection-based methods for: (a) Board 1; (b) Board 2.



(a) Board 3.

(b) Board 4.

Fig. 13: Subset size of two feature-selection-based methods for: (a) Board 3; (b) Board 4.

Next, we compare the two feature selection methods in terms of the size of the reduced syndrome set they provide. In Fig. 12-13, "Subset_M1" refers to the use of complete-case analysis to deal with missing syndromes during feature selection while "Subset_M2" refers to the use of label imputation to address missing values. First, we can see that for both M1 and M2, with an increase in the missing ratio, the size of the extracted syndrome set after feature selection increases first, decreases later, and eventually converges. One possible reason for this phenomenon is that since feature selection is used to extract a set of most informative features for a given board, when the logs contain missing syndromes, then some original informative features may no longer provide useful information; thus feature selection may have to include more alternative features in its extracted subset so that this extracted subset can still give satisfactory diagnosis accuracy. However, feature selection cannot find more appropriate alternative syndromes when the missing ratio is too high. Second, we can see that M2 preserves more syndromes than M1. This is because M2 applies label imputation to deliberately add extra information for missing syndromes while M1 only discards missing syndromes.

Next, in Fig. 14-21, we apply various missing-syndrome handling methods to four different machine learning models: SVM, ANN, Naive Bayes, and Decision Tree, and then compare their diagnosis accuracies obtained for Board 1 and Board 3 under different missing ratios. Note that the "default missing handling methods" in Fig. 14-21 refers to the default missing-syndrome handling method in these four learning models in WEKA as shown in Table. II. In addition, the "feature selection_M1" is the feature selection process that uses complete-case analysis to deal with missing values while "feature selection_M2" applies label imputation to address missing syndromes. The dashed line shown in Fig. 14-21 is the diagnosis accuracy of zero missing ratio, which can be seen as our baseline. The results can be summarized as follows:

First, there is a significant diagnosis accuracy gap between synthetic boards and industrial boards. For example, for the four machine-learning models, i.e., SVM, ANN, Naive Bayes, and Decision Tree, the success ratios for 0% missing ratio for Board 1 are 99.12%, 98.89%, 98.79% and 82.67%, respectively. For Board 3, the baseline success ratios are 78.76%, 77.97%, 81.67% and 70.56%. In the case of Board 4, the baseline success ratios are lower: 72.56%, 70.14%, 71.48% and 61.38% for the four different machine-learning models. This is because industrial boards are more complex than synthetic boards and their log data clearly needs to be augmented to achieve higher success rates. In addition, from Fig. 14-21, we can see that with an increase of missing ratio, the diagnosis accuracy of synthetic boards and industrial boards exhibit a similar decreasing trend for different machine-learning models.

Second, various missing-syndrome-handling methods perform significantly differently for different machine-learning models under different missing ratios. Details are discussed below:

TABLE II: The default technique for handling missing syndromes in the WEKA machine-learning package.

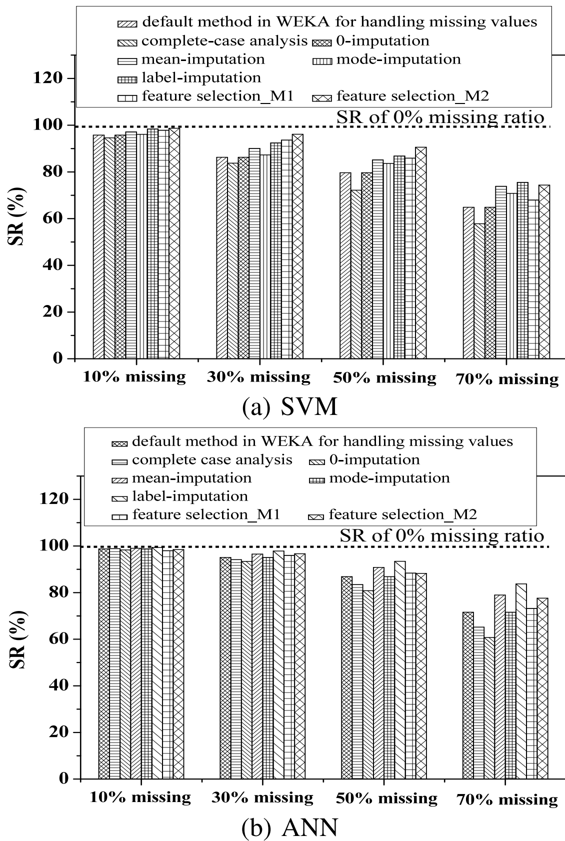
	Default method for Handling missing syndromes
SVM	Mode imputation (discrete features) Mean imputation (continuous features)
ANN	Zero imputation
Naive Bayes	Complete-case analysis
Decision Tree	Fractional instances

1) SVM: Since syndromes in our diagnosis system are denoted as nominal values '0' and '1', the default method for an SVM in our experiments is mode imputation. As shown in figures, the label imputation method always leads to higher diagnosis accuracies for Board 1 (Fig. 14(a)), Board 2 (Fig. 16(a)), Board 3 (Fig. 18(a)) and Board 4 (Fig. 20(a)). The two feature-selection-based methods also perform well, with diagnosis accuracy comparable to mean imputation under different missing ratios. For example, if we consider a 50% missing ratio, the average SRs for Board 1 are 93.42%, 90.79%, 88.39%, 88.24%, 86.84%, 83.47%,80.42% for label imputation, mean imputation, feature selection_M1, feature selection_M2, mode imputation (default in WEKA), complete-case analysis and zero imputation, respectively.
2) ANN: Zero imputation is the default method in ANN's

WEKA implementation. From Fig. 14(b), Fig. 16(b), Fig. 18(b) and Fig. 20(b), we can see that although label imputation is effective, feature selection_M2, which applies label imputation to its feature selection process, performs even better. Taking a 50% missing ratio of Board 1 as an example, feature selection_M2 achieves 89.56% diagnosis accuracy, about 14% higher than the default mechanism and 5% higher than label imputation.

3) Naive Bayes: Complete-case analysis is the default method that WEKA uses with its Nave Bayes classifier. Unlike in other learning models, this method yields impressive results for different missing ratios. Still using the 50% missing ratio case of Board 1 in Fig. 15(a) as our example, the difference between the 93.03% SR of label imputation and the 93.16% SR for complete-case analysis is negligible.

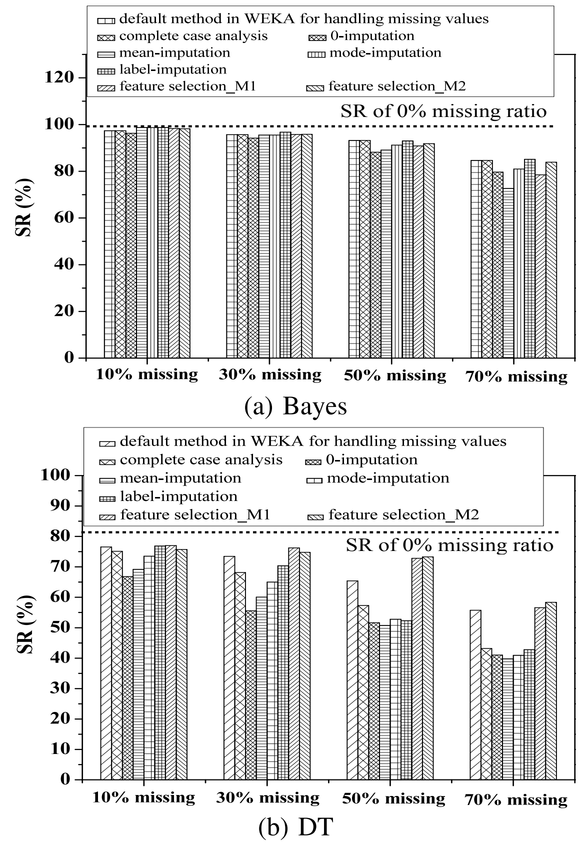
4) Decision Tree: The decision tree model in WEKA, J48, incorporates the fractional instances method to address missing syndromes. As shown in Fig. 15(b), Fig. 17(b), Fig. 19(b) and Fig. 21(b), the most interesting observation here is that all imputation methods (including label imputation) "fail" when the missing ratio is over 50%. In contrast, the two feature-selection-based methods are still effective, and provide better results when the missing ratio is high. For the 50% missing ratio, the diagnosis accuracy of all imputation methods fall to about 50% SR while the feature selection M1 and M2 still maintain their success ratio over 70%.



(a) SVM

(b) ANN

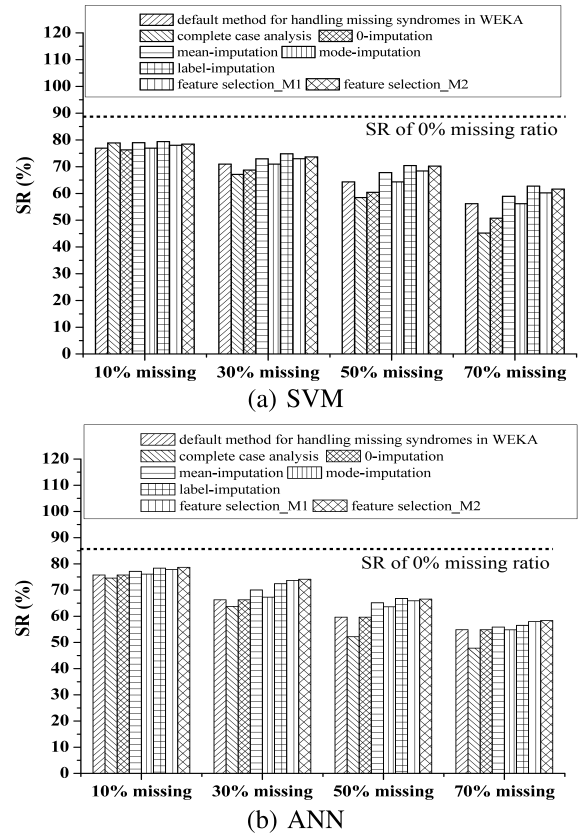
Fig. 14: Comparison of SR of SVM & ANN for Board 1.



(a) Bayes

(b) DT

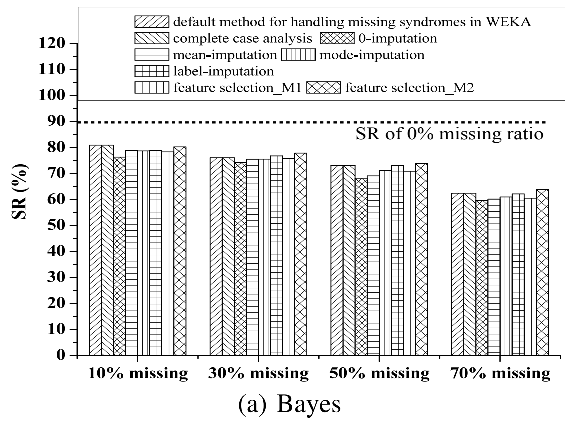
Fig. 15: Comparison of SR of NB & DT for Board 1.



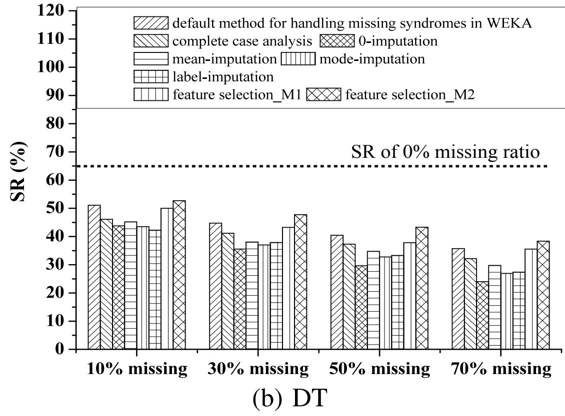
(a) SVM

(b) ANN

Fig. 16: Comparison of SR of SVM & ANN for Board 2.

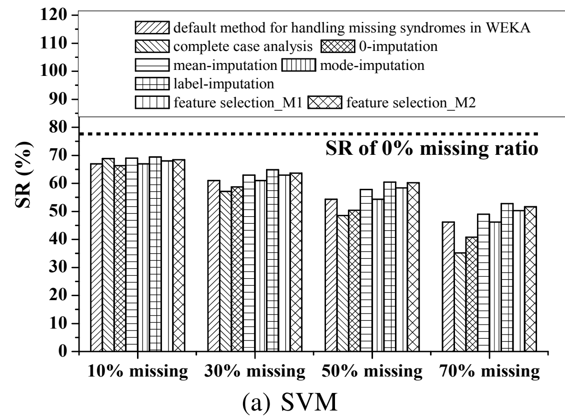


(a) Bayes

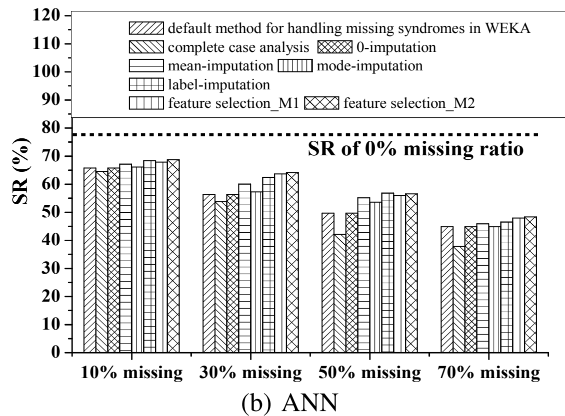


(b) DT

Fig. 17: Comparison of SR of NB & DT for Board 2.

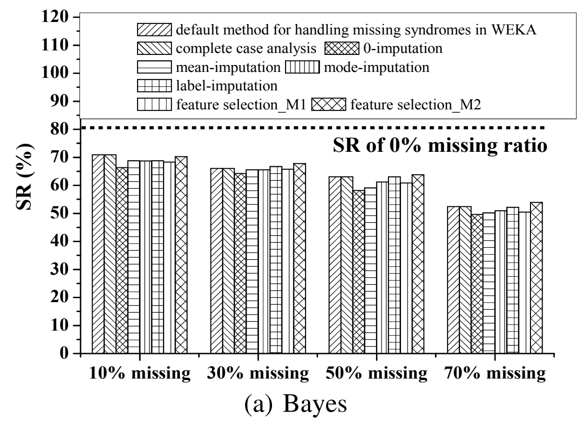


(a) SVM

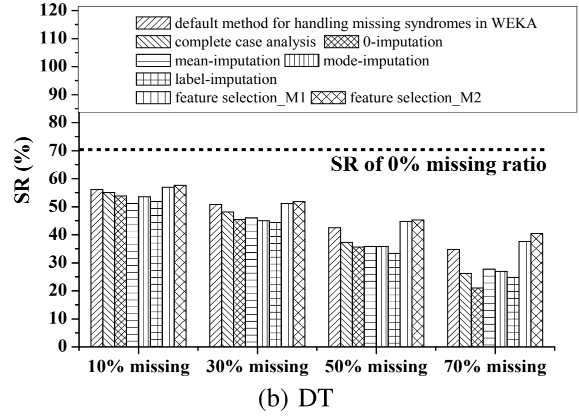


(b) ANN

Fig. 18: Comparison of SR of SVM & ANN for Board 3.

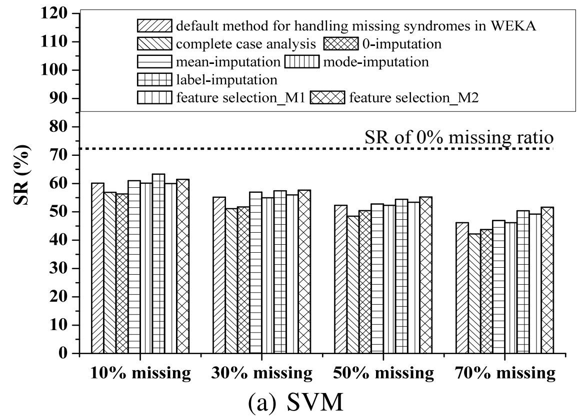


(a) Bayes

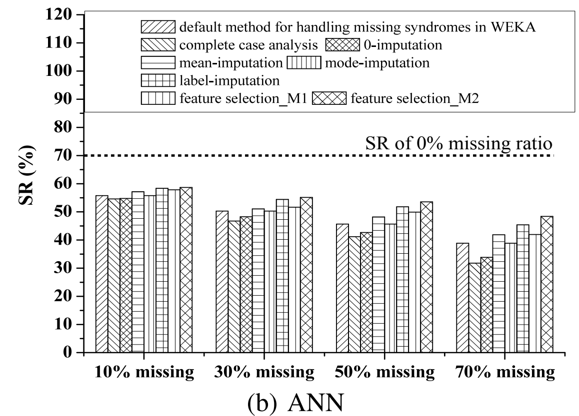


(b) DT

Fig. 19: Comparison of SR of NB & DT for Board 3.

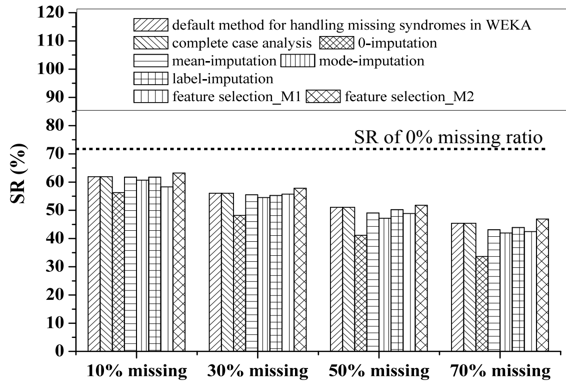


(a) SVM

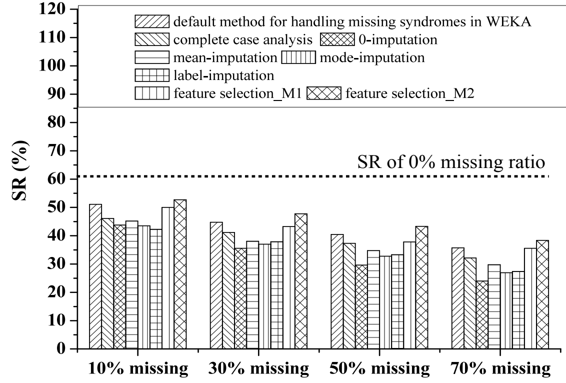


(b) ANN

Fig. 20: Comparison of SR of SVM & ANN for Board 4.



(a) Bayes



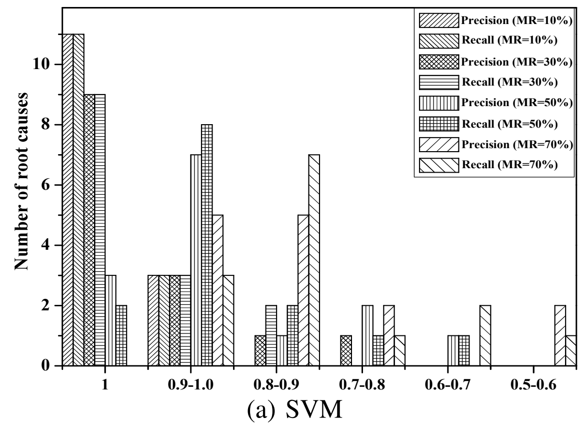
(b) DT

Fig. 21: Comparison of SR of NB & DT for Board 4.

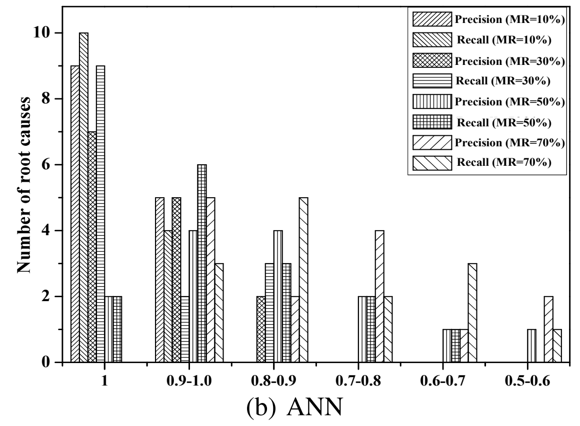
In addition to the success ratio, we also utilize the precision and recall criteria to carry out more fine-grained analysis. We investigate the effect of missing syndromes on root-cause isolation. As shown in Fig. 22-25, when the missing ratio is low, all four machine-learning techniques can identify most root causes without ambiguity, but when the missing ratio is higher, root causes cannot clearly be differentiated from each others.

For example, in Fig. 22, when only 10% of the values are missing, most root causes have precision and recall over 0.8. In contrast, when the missing ratio increases to 70%, the precision and recall of most root causes are distributed between 0.4 and 0.7, dropping significantly from the 10% missing ratio.

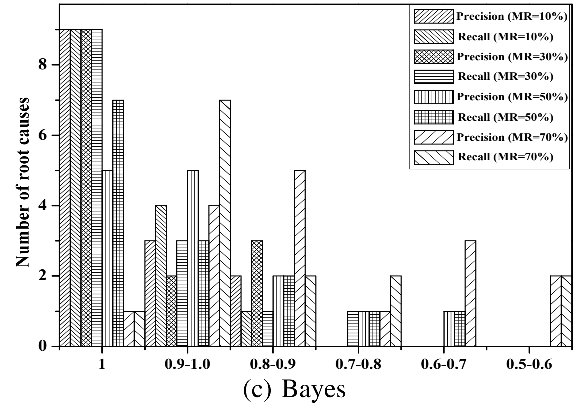
For Board 3, one of the industrial board, Fig. 24 present its precision and recall under different missing ratios. Similar to the two synthetic boards, the number of root causes that have high precision and recall decreases significantly with the increasing of missing ratios. For example, Fig. 24(d) shows that for the DT model, when the missing ratio is 70%, 115 out of 222 root causes have both precision and recall lower than 0.3, which means over half of root causes have become undistinguished. Similar phenomenon can be seen from the other industrial board (Board 4), as shown in Fig. 25. For the DT model, only 32 out of 352 root causes have both precision and recall higher than 0.7 when the missing ratio is 70%.



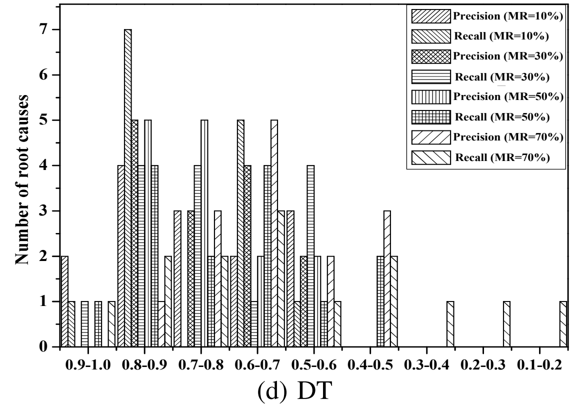
(a) SVM



(b) ANN

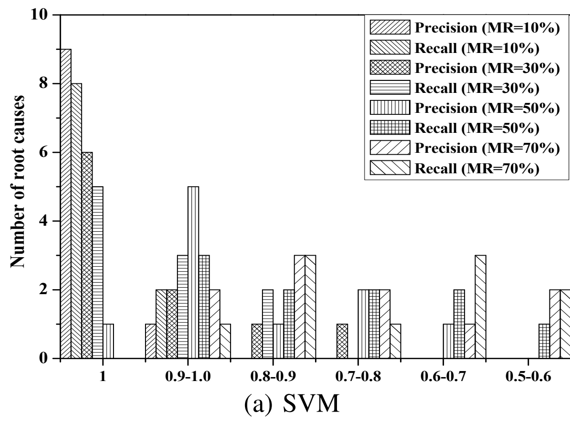


(c) Bayes

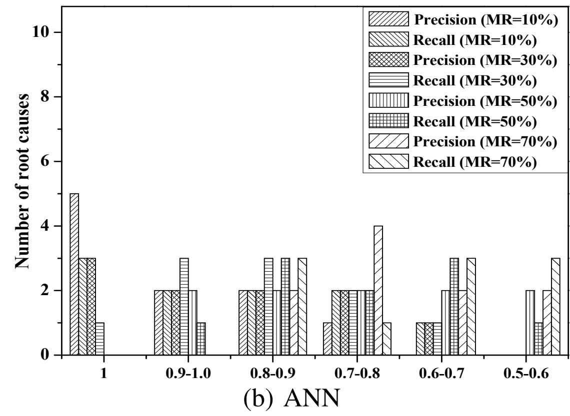


(d) DT

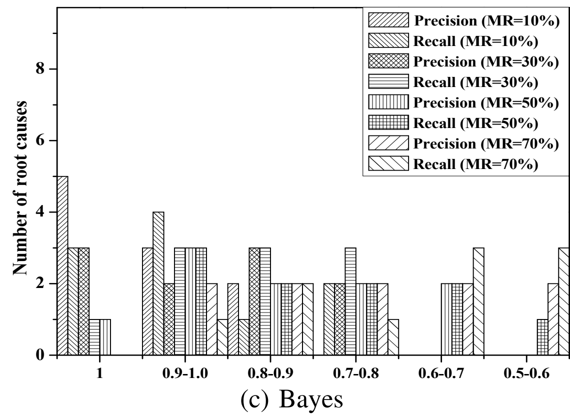
Fig. 22: The precision and recall under various missing ratios for Board 1.



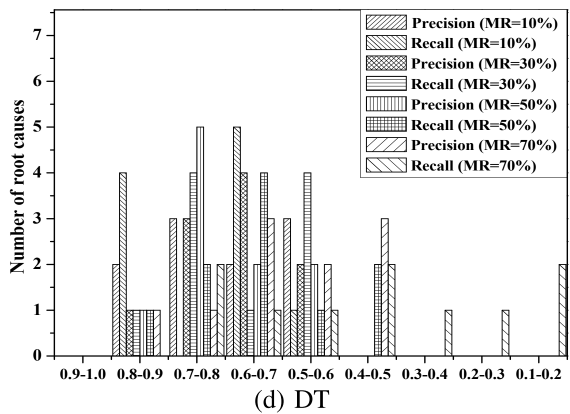
(a) SVM



(b) ANN

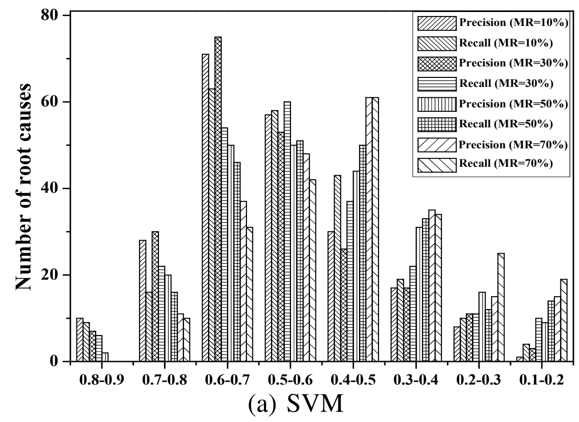


(c) Bayes

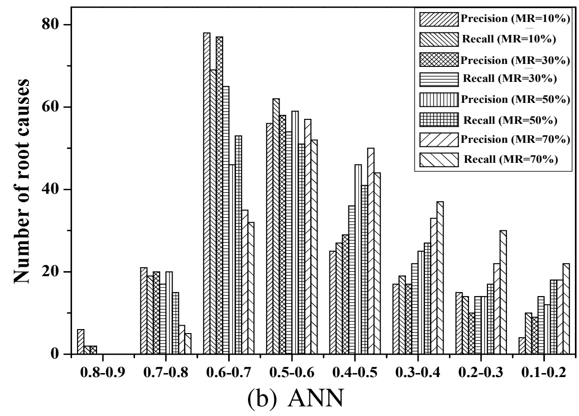


(d) DT

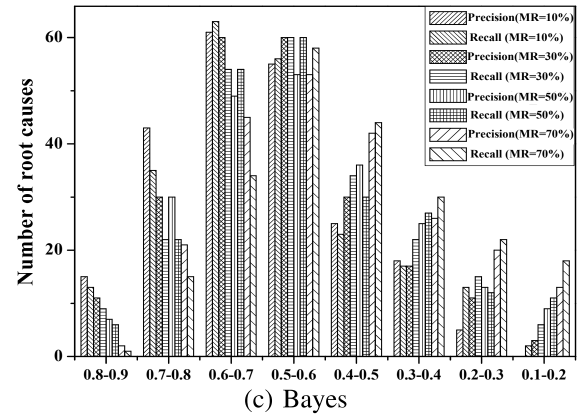
Fig. 23: The precision and recall under various missing ratios for Board 2.



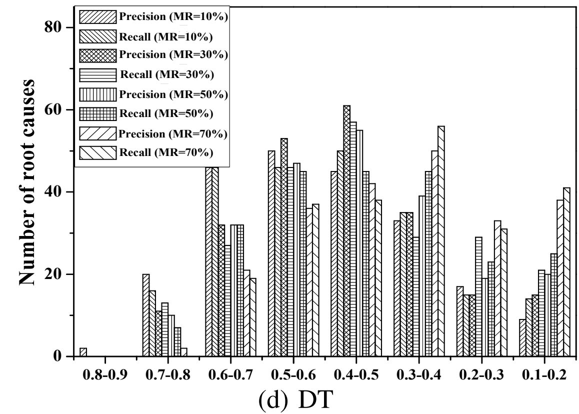
(a) SVM



(b) ANN



(c) Bayes



(d) DT

Fig. 24: The precision and recall under various missing ratios for Board 3.

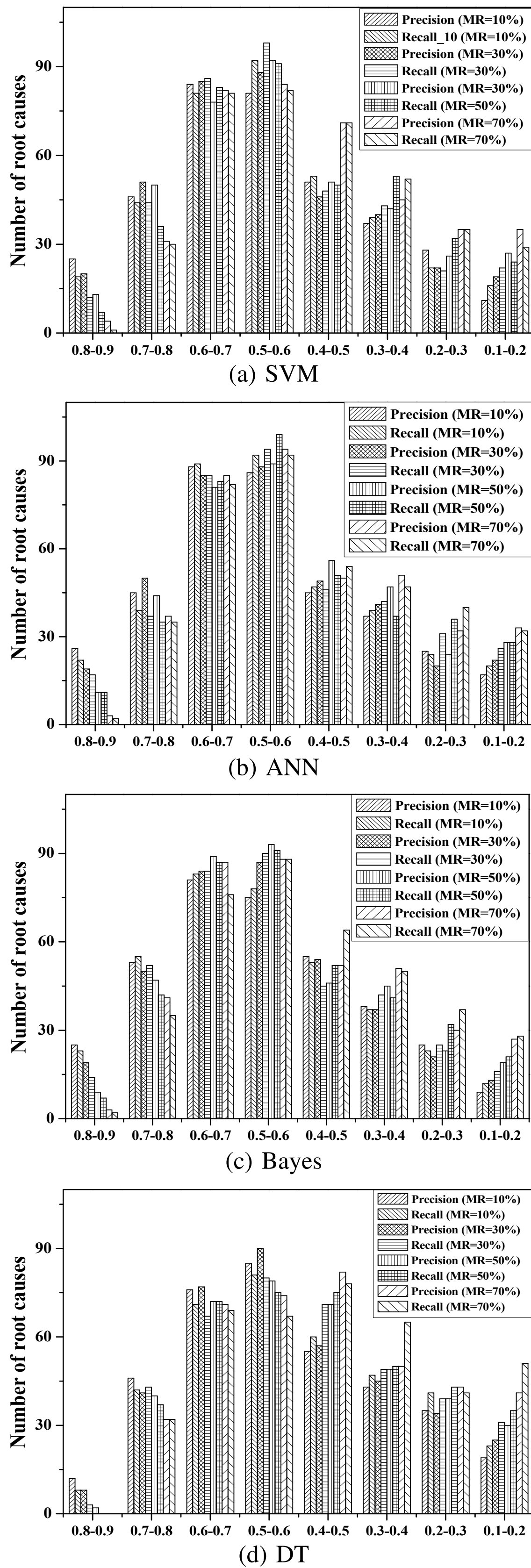
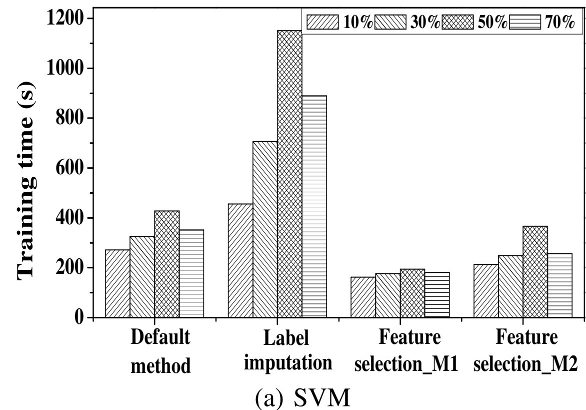


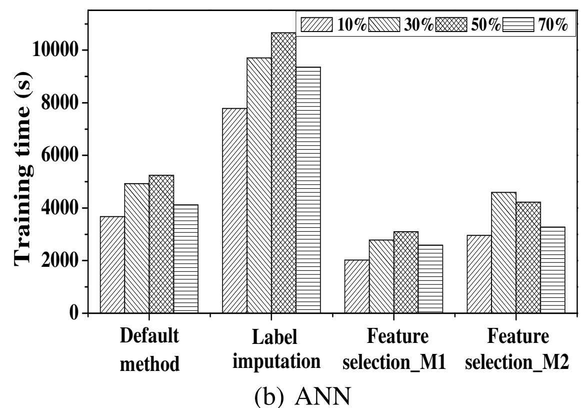
Fig. 25: The precision and recall under various missing ratios for Board 4.

Finally, in Fig. 26-33, we compare the computational complexity of label imputation and the two feature-selection-based methods in terms of diagnosis system training time. The training time for label imputation is much higher than that for the other imputation methods. For synthetic Board 1, the training time of SVM and DT model is an average of 40 minutes, and the ANN model requires over 2 hours. In contrast, the two proposed feature-selection-based methods need only a few minutes for SVM, Bayes, and the DT models, and 50 minutes for the ANN model. The training time of the industrial Board 3 is much higher than that of synthetic Board 1. However, the two proposed feature-selection-based methods still need less training time than other methods. Therefore, the two feature-selection-based methods not only provide comparable diagnosis accuracy as label imputation, but are also computationally more efficient.

In summary, in our experiments with two synthetic boards and two industrial boards, the method "feature-selection-M2" performs better in handling missing syndromes in terms of diagnosis accuracy and training time for most machine learning models. However, since different boards may have significantly different characteristics, we should not conclude that "feature-selection-M2" will always yield the best results. A more realistic conclusion is that the choice of methods to handle missing syndromes depends on two factors:

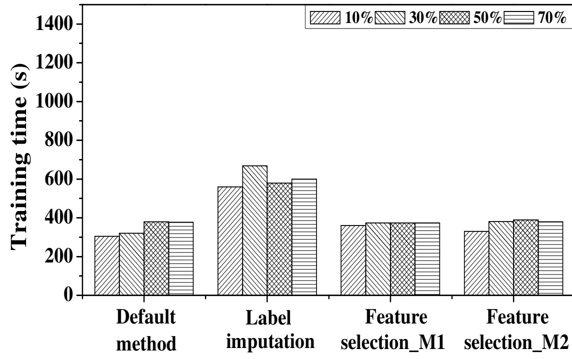


(a) SVM

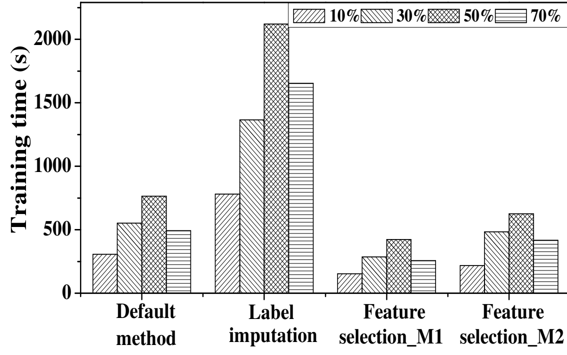


(b) ANN

Fig. 26: Comparison of training time of SVM and ANN models for Board 1.

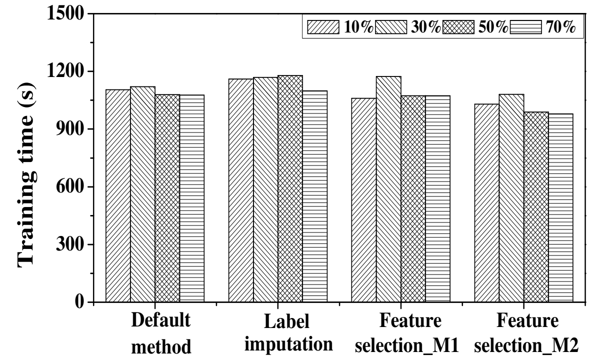


(a) Bayes

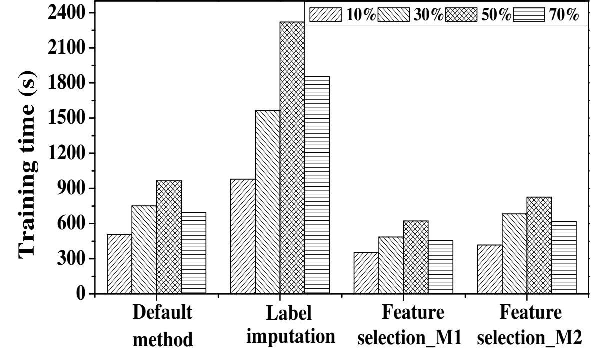


(b) DT

Fig. 27: Comparison of training time of NB and DT models for Board 1.

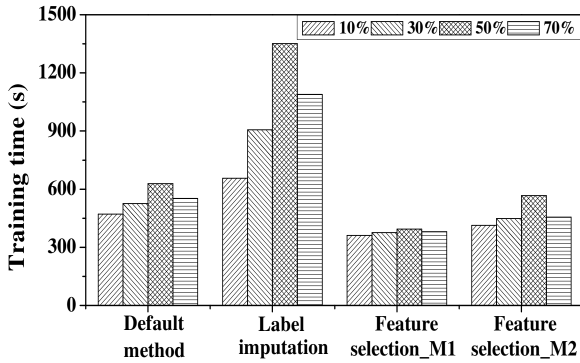


(a) Bayes

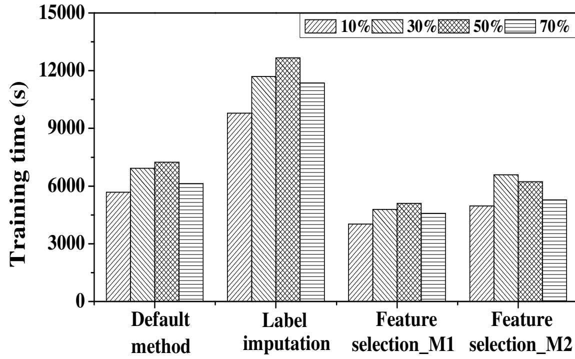


(b) DT

Fig. 29: Comparison of training time of NB and DT models for Board 2.

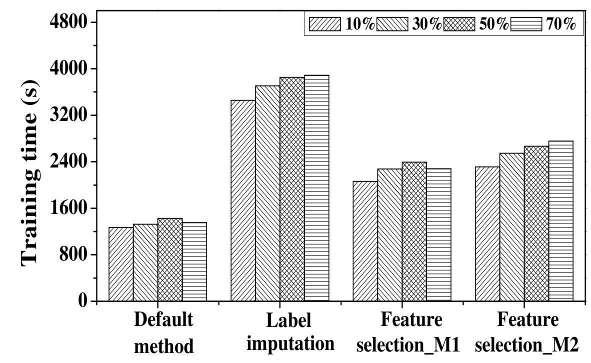


(a) SVM

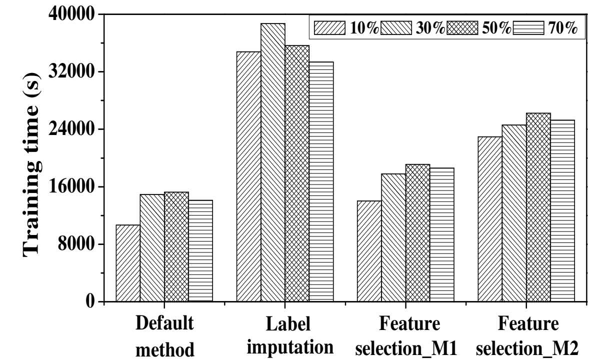


(b) ANN

Fig. 28: Comparison of training time of SVM and ANN models for Board 2.

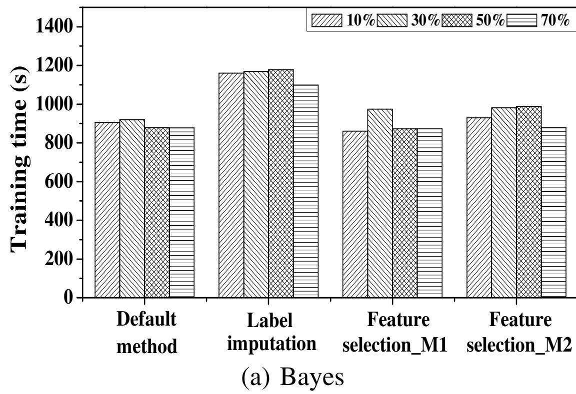


(a) SVM

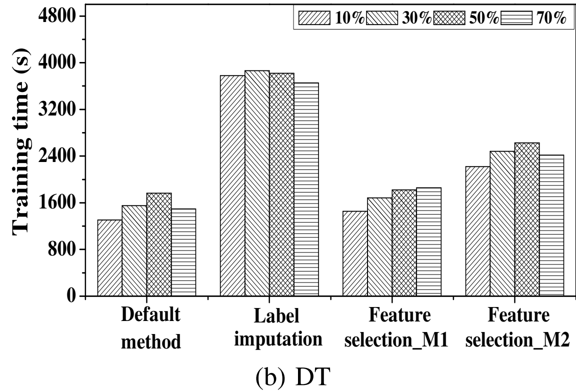


(b) ANN

Fig. 30: Comparison of training time of SVM and ANN models for Board 3.

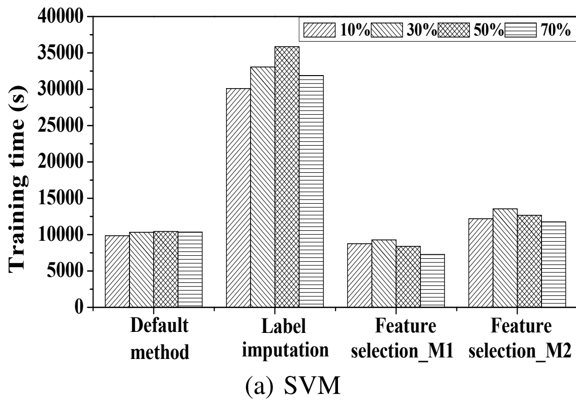


(a) Bayes

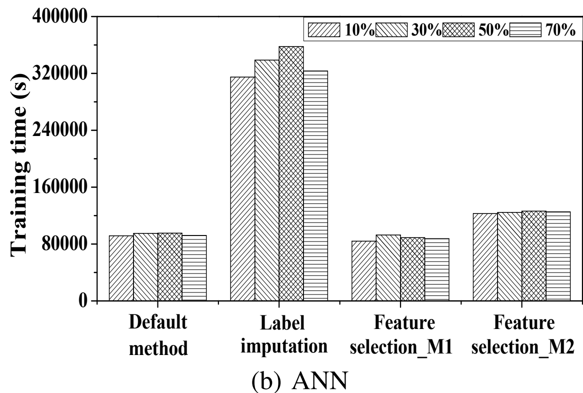


(b) DT

Fig. 31: Comparison of training time of NB and DT models for Board 3.

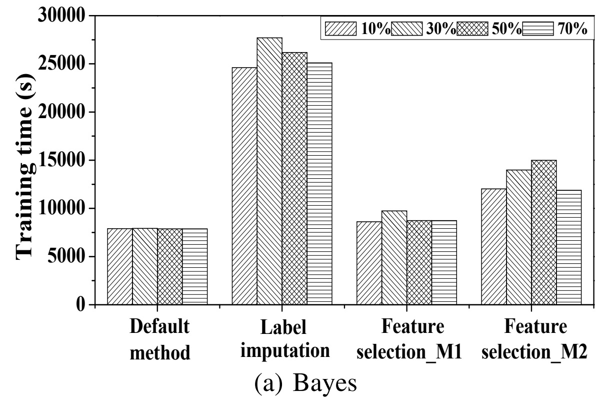


(a) SVM

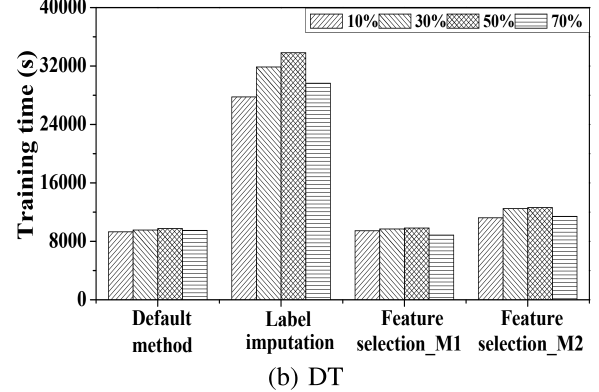


(b) ANN

Fig. 32: Comparison of training time of SVM and ANN models for Board 4.



(a) Bayes



(b) DT

Fig. 33: Comparison of training time of NB and DT models for Board 4.

1) Which machine learning method is used as diagnosis approach: different machine-learning models may prefer different missing-syndrome-handling methods. For example, although label imputation performs best in the ANN model, it "fails" in the DT model.

2) Which objectives are considered: different fault diagnosis system may have different objectives. For example, the goal may be to not only have high diagnosis accuracy, but also to reduce irrelevant and redundant syndromes. In this case, the two methods "feature-selection-M1" and "feature-selection-M2" are more preferable because they have already incorporated the feature selection process when dealing with missing syndromes.

V. Conclusions

We have described the design of a smart board-level diagnosis system that can handle missing syndromes using different methods, among which the label-imputation method and two feature-selection-based methods appear to be the most promising. Two synthetic boards generated from actual board designs currently in the early stage of high-volume production have been used to validate the effectiveness of the proposed method.

References

[1] T. Chakraborty, C.-H. Chiang, and B. Van Treuren, "A practical approach to comprehensive system test & debug using boundary scan based test architecture," in *Proc. ITC*, 2007.

[2] S. Tourangeau and B. Eklow, "Test economics - what can a board/system test engineer do to influence supply operation metrics," in *Proc. ITC*, 2006.
[3] B. Benware *et al.*, "Fault diagnosis with orthogonal compactors in scan-based designs," vol. 27, no. 5, pp. 599–609, 2011.
[4] T. Vo *et al.*, "Design for board and system level structural test and diagnosis," in *Proc. ITC*, 2006.
[5] Z. Conroy *et al.*, "A practical perspective on reducing ASIC NTFs," in *Proc. ITC*, pp. 340–349, 2005.
[6] K. Parker, "Defect coverage of boundary-scan tests: what does it mean when a boundary-scan test passes?," in *Proc. ITC*, pp. 181–189, 2003.
[7] H. Fang *et al.*, "Diagnosis of board-level functional failures under uncertainty using Dempster-Shafer theory," in *IEEE Trans. CAD*, vol. 31, pp. 817–830, 2012.
[8] M. Bushnell, *Essentials of Electronic testing*. Kluwer Academic Publishers, 2000.
[9] W. Fenton, T. McGinnity, and L. Maguire, "Fault diagnosis of electronic systems using intelligent techniques: a review," *IEEE Trans. on Sys., Man., and Cyber., Part C*, vol. 31, no. 3, pp. 269–281, 2001.
[10] D. Manley and B. Eklow, "A model based automated debug process," in *IEEE Board Test Workshop*, pp. 1–7, 2002.
[11] C. O'Farrill, M. Moakil-Chbany, and B. Eklow, "Optimized reasoning-based diagnosis for non-random, board-level, production defects," in *Proc. ITC*, pp. 173–179, 2005.
[12] C. Bolchini, L. Cassano, P. Garza, E. Quintarelli, and F. Salice, "An expert cad flow for incremental functional diagnosis of complex electronic boards," *to appear in IEEE Trans. CAD*, 2015. DOI:10.1109/TCAD.2015.2396997.
[13] C. Bolchini and L. Cassano, "A novel approach to incremental functional diagnosis for complex electronic boards," *to appear in IEEE Transactions on Computers*, 2015. DOI:10.1109/TC.2015.2417537.
[14] K. Huang, H.-G. Stratigopoulos, S. Mir, C. Hora, Y. Xing, and B. Kruseman, "Diagnosis of local spot defects in analog circuits," *IEEE Transactions on Instrumentation and Measurement (TIM)*, vol. 61, no. 10, pp. 2701–2712, 2012.
[15] F. Ye *et al.*, "Board-level functional fault diagnosis using multi-kernel support vector machines and incremental learning," *IEEE Trans. CAD*, vol. 32, no. 5, pp. 723–736, 2013.
[16] F. Ye *et al.*, "Adaptive board-level functional fault diagnosis using decision trees," in *Proc. ATS*, pp. 202–207, 2012.
[17] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I. H. Witten, "The weka data mining software: An update," *SIGKDD Explor. Newsl.*, vol. 11, no. 1, pp. 10–18, 2009.
[18] Z. Zhang *et al.*, "Diagnostic system based on support-vector machines for board-level functional diagnosis," in *Proc. ETS*, pp. 170–175, 2012.
[19] Z. Zhang *et al.*, "Smart diagnosis: Efficient board-level diagnosis and repair using artificial neural networks," in *Proc. ITC*, 2011.
[20] L. Amati *et al.*, "An incremental approach to functional diagnosis," in *Proc. DFT*, pp. 392–400, 2009.
[21] K. Pelckmans *et al.*, "Handling missing values in support vector machine classifiers," *Neural Networks*, vol. 18, no. 5, pp. 684–692, 2005.
[22] G. Chechik *et al.*, "Max-margin classification of data with absent features," *The Journal of Machine Learning Research*, 2008.
[23] I. Rish, "An empirical study of the naive bayes classifier," in *IJCAI 2001 workshop on empirical methods in artificial intelligence*, vol. 3, pp. 41–46, IBM New York, 2001.
[24] J. Quinlan, "Induction of decision trees," *Machine Learning*, vol. 1, no. 1, pp. 81–106, 1986.
[25] M. Saar-Tsechansky and F. Provost, "Handling missing values when applying classification models," *Journal of machine learning research*, 2007.
[26] B. Prieto, J. D. L. Asian, and D. Maravall, *Reconfigurable Hardware Implementation of Neural Networks for Humanoid Locomotion*. 2005.
[27] Y. Ding, X. Peng, and X. Fu, *The Research of Artificial Neural Network on Negative Correlation Learning*. 2009.
[28] V. Vapnik, "An overview of statistical learning theory," *IEEE Transactions on Neural Networks*, vol. 10, no. 5, pp. 988–999, 1999.
[29] C.-C. Chang and C.-J. Lin, "LIBSVM: A library for support vector machines," Software available at http://www.csie.ntu.edu.tw/cjlin/libsvm.
[30] C. Cortes and V. Vapnik, "Support-vector networks," *Journal of Machine Learning*, vol. 20, pp. 273–297, 1995.