Board-level functional fault diagnosis using data mining

C. Bolchini · P. Garza · E. Quintarelli

Abstract This paper presents an approach for performing functional diagnosis of complex systems by means of data mining. The technique allows to derive a set of rules from a functional model of the system for efficiently driving the diagnosis procedure towards the identification of the most promising faulty candidate. The approach is adopted within an incremental method, to limit the number of tests to be performed, thus reducing costs and effort.

Keywords Functional Diagnosis · Data Mining · Faulty candidates

1 Introduction

Reasoning-based methods and machine-learning techniques are receiving a lot of attention as a possible contribution for performing an effective functional fault diagnosis. In fact, when dealing with complex boards, functional diagnosis is the only affordable approach, even if several issues need to be dealt with, when defining a relevant approach that allows to identify the most probable faulty (sub)component with a high confidence by performing a small number of tests (to reduce the overall time and effort). In the past, several solutions have been defined, among which we mention rule-based ones [4], that adopt rules stated as “if test output(s) → faulty component”. Other approaches are based on a model of the system under consideration [5], possibly adding some reasoning-awed knowledge ([7]), requiring a good understanding and information of the relationship between the components and the tests. More recently, reasoning-based solutions have been presented, exploiting Bayesian Networks ([2,12]), Decision Trees and Support-Vector Machines ([10,11]). An analysis of the benefits and limitations of the application of different machine learning techniques to test data collection for functional diagnosis is presented in [8], to compare the solutions aimed at limiting the amount of tests being executed, especially if they do not add significant information to either speed up the diagnosis or to increase the accuracy. In general, two are the main issues when trying to improve functional fault diagnosis applied at high abstraction level: i) reduce the number of tests to be executed to identify the faulty candidate, instead of collecting the complete failure responses and ii) quantify the confidence in the performed diagnosis.

Several statistical learning techniques have been explored in the literature to deal with these two issues, but, to the best of our knowledge, the use of data mining (DM) [6] has not been investigated to this purpose. DM research area focuses on studying algorithms and techniques to find interesting patterns representing implicit knowledge stored in massive data repositories; it has been applied to different fields, but is rapidly receiving interest to improve the quality of the manufacturing process. In this paper we apply DM algorithms for inferring correlations among data in the form of association rules [1,3,6]. The correlations are extracted from test vectors to rapidly find out relationships between test vectors and the fault component they actually detect.

The structure of the paper is as follows: the next section introduces the basic concepts related to functional diagnosis (i. e., the adopted system model, partial syndromes), also presenting a running example. The proposed approach is discussed in Section 3, by referring to the adopted case study. A preliminary evaluation of the achieved results is
presented in Section 4 together with considerations on ongoing and future work.

2 Background

When performing diagnosis, a set of tests \( T = \{T_1, T_2, \ldots, T_n\} \) is available that can be executed to identify the faulty component (e.g., memory, processor, FPGA, . . . ) among the ones constituting the system under test, \( C = \{C_1, \ldots, C_m\} \). A traditional approach consists in applying the entire set of tests \( T \) and collect the set of corresponding outcomes, classified as either PASS or FAIL, called syndrome. The syndrome is exploited by the diagnosis process, by referring to a model that puts into relation the tests and the components, that is the system model. The approach can work at different abstraction levels, where components can be processors, RTL modules or single gates, based on the available test sets, and the problem under consideration.

2.1 The system model

In this work we adopt the system model proposed in [2], dubbed Components-Tests Matrix – CTM, where each entry \( CTM_{ij} \) qualitatively or quantitatively specifies how probable it is that test \( T_j \) fails when component \( C_i \) is faulty. A simplified qualitative scale is used (high, Medium or Low), considering the difficulty of computing an accurate value and benefiting from the ability of probabilistic reasoning systems to tolerate imperfection. These qualitative values expressed by a test engineer providing the system description are then automatically converted in coarse grain values, to be used in reasoning systems. More precisely, the adopted scale is \( H = 0.9, M = 0.5 \) and \( L = 0.1 \). An example of the CTM is reported in Fig. 1, used as a running example.

2.2 Running example

In this running example, a simple circuit consisting of 3 components is considered, having 5 tests available. As an example of the meaning of each entry, should component \( C_1 \) fail, test \( T_2 \) might pass or fail, and the same is true for \( T_5 \).

![CTM used as a running example](image)

<table>
<thead>
<tr>
<th>Components</th>
<th>( T_1 )</th>
<th>( T_2 )</th>
<th>( T_3 )</th>
<th>( T_4 )</th>
<th>( T_5 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( C_1 )</td>
<td>( M )</td>
<td>( M )</td>
<td>( M )</td>
<td>( M )</td>
<td>( M )</td>
</tr>
<tr>
<td>( C_2 )</td>
<td>( H )</td>
<td>( M )</td>
<td>( L )</td>
<td>( L )</td>
<td>( L )</td>
</tr>
<tr>
<td>( C_3 )</td>
<td>( M )</td>
<td>( L )</td>
<td>( M )</td>
<td>( M )</td>
<td>( L )</td>
</tr>
</tbody>
</table>

1: \textbf{procedure} DIAGNOSIS(CTM) \triangleright \text{Uses the system model}
2: \( RS \leftarrow \text{extractRules}(CTM) \) \triangleright \text{Rule Set extracted from CTM}
3: \( FC \leftarrow \emptyset \) \triangleright \text{Set of Not Faulty Candidate(s)}
4: \( NFC \leftarrow \emptyset \) \triangleright \text{Partial Syndrome - no tests executed}
5: \( PS \leftarrow \emptyset \) \triangleright \text{Nothing else to be done or FC identified}
6: \textbf{while} \( RS \neq \emptyset \) AND \( FC = \emptyset \) \textbf{do} \triangleright \text{Additional approach consists in applying the entire set of tests}
7: \( R_i \leftarrow \text{SelectTopRankingRule}(RS) \)
8: \( PS_i \leftarrow \text{ApplyTestForRule}(R_i) \) \triangleright \text{Partial syndrome - no additional test}
9: \textbf{if} \( R_i \) is satisfied \textbf{then}
10: \( FC \leftarrow \text{Consequent Rule} \)
11: \textbf{else}
12: \( RS \leftarrow \text{updateRules}(PS_i) \) \triangleright \text{Prune not applicable rules}
13: \( NFC \leftarrow \text{updateNotFaultyCompSet}(PS_i, CTM) \)
14: \( RS \leftarrow \text{updateRules}(NFC) \) \triangleright \text{Prune rules w.r.t. NFC}
15: \textbf{end if}
16: \textbf{end while}
17: \textbf{return} \( FC \) \triangleright \text{Faulty component or no diagnosis}
18: \textbf{end procedure}

![Pseudo-code of the data mining-based diagnosis process](image)

2.3 Incremental diagnosis process

Rather than running all tests and identifying the possible faulty candidate based on the entire syndrome (e.g., [10, 11]), the authors in [2] propose to execute only some of all tests, and to determine whether the partial syndrome can suffice. If not, additional tests are executed and the new (partial) syndrome is exploited. The aim is to reduce the number of tests to be executed, which is a critical aspect in terms of diagnosis costs (time and effort). At each step, thus, the method must define how the next test to be executed is selected, and when to stop, either because a faulty component has been identified (with a satisfactory level of confidence) or because additional tests would not add any meaningful information.

3 The proposed approach based on data mining

The incremental approach is here implemented by exploiting data mining, aiming at extracting further knowledge from the system model with respect to that bayesian solution presented in [2], for a more efficient diagnostic procedure.

More precisely, from the CTM an initial model composed of association rules is extracted. The incremental method, at each step, selects the most promising test to be executed, and based on the partial syndrome and the exploited association rule it determines whether there is a probable faulty candidate, or additional tests need to be executed. Fig. 2 reports the pseudo-code of the method, whose detailed aspects are presented in the next paragraphs.

3.1 Rules extraction and ranking

Association rules [1] describe the co-occurrence of data items and are usually represented as implications in the \( X \Rightarrow Y \)
form, where \( X \) and \( Y \) are two arbitrary sets of data items such that \( X \cap Y = \emptyset \). For instance, a rule extracted from the running example \( \text{CTM} \) is \( \{ T_2 \} \Rightarrow C_1 \), stating that if both \( T_2 \) and \( T_4 \) fail, then \( C_3 \) is the faulty component.

The quality of an association rule is evaluated by means of support and confidence measures. Support corresponds to the frequency of the set \( X \cup Y \) in the dataset; confidence corresponds to the conditional probability of finding \( Y \) having found \( X \), and is given by \( \frac{\text{sup}(X \cup Y)}{\text{sup}(X)} \). In this paper, the set \( X \) is composed of faulty tests and \( Y \) is the predicted faulty component. For instance, the rule \( \{ T_2 \} \Rightarrow C_3 \), with confidence equal to 100%, states that if both \( T_2 \) and \( T_4 \) fail then the faulty component is \( C_3 \) with a 100% estimated probability.

The mentioned association rules are mined from the \( \text{CTM} \) representing the system under analysis. However, to take into consideration the faulty probability reported in the considered \( \text{CTM} \), we exploit a special type of association rules called weighted association rules [9]. The main difference between weighted association rules (WARs) and traditional ones is given by the fact that WARs considers also the importance of each item in the analyzed data. In particular, each item in the input data is associated with a weight representing its importance. The assigned weights are used to compute a weighted version of the support measure where the frequency of a rule and the weights of its items are combined. In this paper, we use the quantitative relationship between \( C_i \) and \( T_j \) expressed in the \( \text{CTM} \) to assign an appropriate weight to each item (using the 0.9, 0.5, 0.1 scale).

The weighted association rule mining process is divided into two subtasks: (i) find all the sets of items (itemsets) whose weighted support exceeds a given threshold \( \text{minsup} \) and (ii) generate, starting from the mined itemsets, the rules with a confidence greater than a specified threshold \( \text{minconf} \). Fig. 3 reports some of the rules mined from the running example \( \text{CTM} \) in Fig. 1.

A ranking procedure is applied on the mined rule set in order to identify the “best” predictive rules. In particular, a sort order based on confidence, rule length, and support is imposed on the mined rule set \( RS \). The rule length of a rule is defined as the number of items (tests) in the antecedent of the considered rule. The highest quality rules are those characterized by a high confidence.\(^1\) Hence, we sort rules based on confidence. When two rules have the same confidence, the shortest one is preferred, to limit the number of test to be performed. Finally, when two rules have the same confidence and length values, the weighted support is considered.

The ranked rule set represents, combined with the \( \text{CTM} \), the model exploited by our approach to select the subset of tests to be executed and predict the faulty component as it is described in the following paragraphs.

\(^1\) We recall that the confidence value represents an estimate of the conditional probability that given the tests in the antecedent of the rule the faulty component is the one in the consequent of the rule.

### Fig. 3 Rule set mined from the sample \( \text{CTM} \) reported in Fig. 1

<table>
<thead>
<tr>
<th>Rule</th>
<th>Confidence</th>
<th>Weighted Support</th>
</tr>
</thead>
<tbody>
<tr>
<td>{ } \Rightarrow C_2</td>
<td>100%</td>
<td>90%</td>
</tr>
<tr>
<td>{ T_1 } \Rightarrow C_3</td>
<td>100%</td>
<td>50%</td>
</tr>
<tr>
<td>{ T_2 } \Rightarrow C_3</td>
<td>100%</td>
<td>10%</td>
</tr>
<tr>
<td>{ C_2 } \Rightarrow C_3</td>
<td>100%</td>
<td>10%</td>
</tr>
<tr>
<td>{ C_3 } \Rightarrow C_3</td>
<td>64.3%</td>
<td>90%</td>
</tr>
<tr>
<td>{ } \Rightarrow C_3</td>
<td>50%</td>
<td>50%</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>

3.2 Next test selection

To select the next test to be performed, we consider the mined rules in the sort order presented above. The first rule \( R_3 \) in the sort order is selected (see Fig. 2, line (7)) and the tests in its antecedent are considered, one at a time, and at each outcome the partial syndrome is updated. If all the tests in \( R_3 \) fail, it means that \( R_3 \) can be applied and the component in the consequent of \( R_3 \) is selected as the faulty component. Otherwise, other tests are needed. However, before considering the next rule in the sort order, the rule set is pruned and a set of (surely) not faulty components is identified based on the partial syndrome. These pruning procedures allow avoiding the execution of useless tests and limiting the number of components under analysis.

3.3 Not faulty candidate identification

The current partial syndrome \( \text{PS} \) can be used to prune part of the search space. In particular, some components can be excluded from the set of candidate faulty components. Consider the first rule in Fig. 3. Based on it, the next test to be applied is \( T_1 \). Suppose that \( T_1 \) passes. It means that we cannot apply the current rule and we need to execute more tests in order to identify the faulty component. Moreover, the information about the output of \( T_1 \) allows our approach to prune some components. By considering the running example \( \text{CTM} \) and the fact that the output of \( T_1 \) is equal to pass, we can predict that \( C_2 \) cannot be the faulty component because the probability of a failure of \( T_1 \) when \( C_2 \) is broken is high. Based on this consideration, \( C_2 \) is included in the set of not faulty components (see Fig. 2, line (13)) and the rules with \( C_2 \) as consequent are considered not correct in this current context and are pruned (see Fig. 2, line (14)). Fig. 4 reports the available rules after pruning. The applied pruning allows reducing the number of rules and potentially the number of performed tests.

3.4 Rules update

Based on the available partial syndrome \( \text{PS} \), the rule set can be further pruned, given that some rules are not applicable
The results are comparable in terms of the executed number of tests (which was the adopted metric in order to promote the incremental strategy). This is an encouraging result, since the potential of the data mining approach has yet to be fully exploited with respect to the following issues: i) the use of an initial test set, that proved to be a fundamental aspect for the Bayesian engine, ii) the analysis of the system model, to purge possibly elements that add little information and increase the size of the problem, iii) possibility to leverage on the confidence of the result for a shorter diagnostic process.

In general, the approach is suitable for board-level scenarios and data mining promises to offer several means to improve and tune the diagnosis.

### References