

A Configurable Board-level Adaptive Incremental Diagnosis Technique based on Decision Trees

Cristiana Bolchini, Luca Cassano

Dipartimento di Elettronica, Informazione e Bioingegneria - Politecnico di Milano, Italy

Email: {cristiana.bolchini, luca.cassano}@polimi.it

Abstract—Functional diagnosis for complex electronic boards is a time-consuming task that requires big expertise to the diagnosis engineers. In this paper we propose a new engine for board-level adaptive incremental functional diagnosis based on decision trees. The engine incrementally selects the tests that have to be executed and based on the test outcomes it automatically stops the diagnosis as soon as one or more faulty candidates can be identified, thus allowing to reduce the number of executed tests. Moreover, we propose a configurable early stop condition for the engine that allows to further reduce the number of executed tests leveraging the diagnosis accuracy. The effectiveness of the proposed approach has been assessed using a set of synthetic but realistic boards and three industrial boards.

Keywords—Adaptive Diagnosis; Decision Trees; Functional Diagnosis; Incremental Diagnosis; Machine Learning.

I. INTRODUCTION AND RELATED WORK

The complexity of modern electronic systems might make structural test/diagnosis approaches not suitable, therefore functional ones have been designed to overcome such limitation [1]. Functional diagnosis aims at identifying the cause of a failure of an electronic system, i.e., a mismatch between the expected behaviour and the actual one, by applying a set of input stimuli and by observing the system responses, without knowing the internal structure of the system under analysis.

Several motivations may lead companies to apply functional diagnosis techniques: 1) the identification of the faulty component for possible repair in case of an expensive board, hosting several subsystems; 2) the identification of the unfaulty components for possible reuse in case of an unfixable board hosting expensive subsystems; and 3) the collection of information on components' failure rates, to be able to monitor the quality of suppliers' products. Functional diagnosis is a very complex task; indeed, the effectiveness of diagnosis activities is often more affected by the expertise and the skills of the test/diagnosis engineers than by the CAD tools mainly because the information provided by the currently available software tools often lack of preciseness. As a result, a number of machine learning-based "intelligent" techniques have been proposed in the last years to assist the operator.

The main problem addressed by most of the available techniques, such as the ones presented in [2], [3], [4], [5], is the extraction of an effective system model starting from a limited amount of data belonging to previously performed testing/diagnosis activities. The extracted model is then used in a "traditional" way, i.e., by applying all the available tests and collecting all outcomes (referred to as a *complete syndrome*). Such solutions are thus meant to be used at design time, while not aimed at driving the actual runtime diagnosis process.

Given a model of the board under analysis, a new strategy aiming at reducing the number of tests to be executed to achieve the diagnosis has been introduced in [6], dubbed *adaptive, incremental functional diagnosis*. The approach, based on Bayesian Belief Networks, incrementally executes (groups of) tests and, based on their outcomes, constituting a *partial syndrome*, adapts the execution order of the remaining tests and interrupts the process as soon as the faulty component can be identified. Subsequently, Data Mining has been adopted as an alternative engine to support the methodology ([7]), offering interesting improvements. A comparative analysis of different machine learning-based engines has preliminarily shown that decision trees constitute a good candidate ([8]).

In this paper, we propose a refined and tuned exploitation of decision trees, for adaptive incremental functional diagnosis, introducing also a customizable feature to leverage accuracy in favour of a further reduction in the number of executed tests (thus lowering time and cost of the diagnosis process). In particular, the proposed technique (referred to as an *early stop condition*) is configurable, allowing the user to specify the maximum acceptable error (in terms of fault-free components erroneously identified as faulty), thus bounding the accuracy reduction.

The remainder of this paper is organized as follows: Section II discusses the problem we face and presents the decision trees; Section III shows the proposed approach; Section IV presents the early stop mechanism; Section V presents the metrics defined for the evaluation of the method and discusses the experimental results; Section VI concludes the paper.

II. PROBLEM STATEMENT AND BACKGROUND

A. Problem Statement

Let us consider an electronic board that has exhibited an erroneous behaviour, and a stakeholder interested in identifying the faulty component(s). An adaptive incremental functional diagnosis approach performs a test *at a time* and determines (*incrementality*) whether enough information is available to determine the diagnosis, or more test outcomes are necessary. At each iteration, the test to be executed is selected based on the previous collected information (*adaptability*); as a consequence, the sequence of tests is *dynamically determined*. More in detail, the steps iteratively performed are: 1) select a test to be executed on the faulty board; 2) execute and collect the outcome; 3) based on the current *partial syndrome* decide either to stop the diagnosis process pointing to the faulty component or to execute a new test, restarting from step 1.

We refer to *one test* although it generally consists of a short test suite verifying a functionality of the component under consideration, corresponding to a set of input patterns and/or probing activities. The fundamental characteristic of the *test* as referred to is that it is the smallest unit of activity the test engineer has defined.

Adaptive incremental diagnosis approaches aim at “identifying the faulty component(s) with the highest *accuracy* while executing the smallest *number of tests*”. We refer to accuracy as the ability of the approach to identify the faulty candidate component(s) introducing neither false positives, i.e., fault-free components erroneously considered as faulty, nor false negatives, i.e., faulty components erroneously considered as fault-free. We refer the reader to Section V-A for a formal definition of the accuracy and number of test metrics.

Finally, it is worth noting that in the context of adaptive incremental diagnosis two assumptions are fundamental: 1) At least a faulty component exists, since diagnosis takes place *because* a failure has been observed; and 2) only one faulty component exists, otherwise the identification of the source of a problem could not imply the process can be interrupted.

B. Decision Trees

Decision Trees (DTs) are decision-support tree-like structures used to classify instances of a context of objects based on the values of a set of features [9]. Generally speaking, the internal nodes of a DT represent *observations* of the features; the branches represent decisions taken according to the observations and the leaves represent class labels, i.e., decisions taken after observing all/some features. Each path from the root to a leaf represents a *classification rule*. More in details, each internal node is associated with a feature and with a *decision rule*, i.e., the rule that determines which branch has to be taken based on the observation of the feature. Two kinds of features (and associated decision rules) exist: *ordered*, i.e., the actual value of the feature is compared with a threshold, and *categorical*, i.e., the actual value of the feature has to fall into a given subset. Thus in each internal node a <feature, decision rule> pair is stored, referred to as a *split*.

DTs are built recursively, starting from the root node. According to the available *training data*, i.e., vectors of sample values of all the features and the corresponding class, for each node a split is found. Then, the remaining training samples are divided by using the split among the children of the node and again the children nodes are recursively split. At each node, this procedure may stop in one of the following cases: 1) the depth of the constructed tree branch has reached a specified maximum value; 2) the number of training samples in the node is lower than a threshold and thus it is not significant to further split the node; 3) all the samples in the node belong to the same class; and 4) the best identified split does not provide any improvement compared to a random choice. Once a DT has been built, instances of the problem can be classified at *run time* by starting from the root and traversing the tree based on the actual values of the features obtained through observations.

In the present context, the DT is used to *classify* the faulty component(s) associated with the outcomes of tests (observations), either PASS or FAIL; the leaves of the tree are the faulty candidate(s). The proposed approach aims at

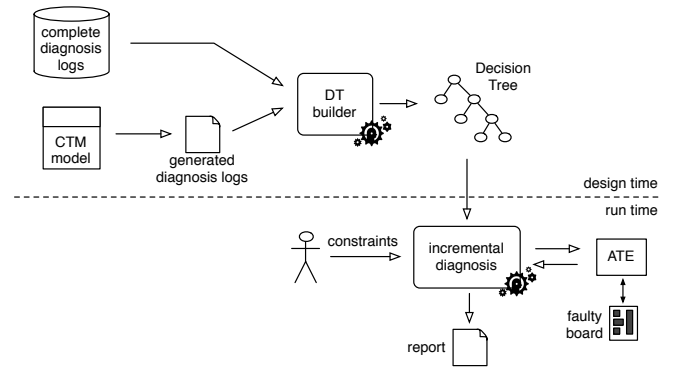


Fig. 1. The proposed incremental diagnosis flow using Decision Trees

building a DT starting from the information gathered from previous diagnosis sessions, or from a system model provided by the test/diagnosis engineers and then – at runtime – by traversing it and executing the corresponding tests to perform the diagnosis. The next section introduces the preliminary, design time, activity for the creation of the DT, as well as its runtime exploitation for the adaptive incremental diagnosis.

III. THE PROPOSED DIAGNOSIS APPROACH

The proposed approach consists of two phases, a *design time* one, and a *run time* one, as shown in Figure 1. At design time, the decision tree, which represents the key element of the proposed adaptive incremental diagnosis, is built. This phase, although complex in relation with the amount of available data, is performed off-line at the beginning and can be periodically recalled, to tune or adjust the DT with respect to new recent diagnoses. The DT is then fed to the incremental engine, that at run time, when a failing board needs to be analysed, iteratively and adaptively requests the execution of tests and collects the outcomes from an Automatic Test Equipment (ATE) until the diagnosis can be performed.

The decision tree represents the information putting into relation failing/passing tests with the faulty components. To build such a tree, it is necessary to have a *log* containing a number of <complete syndrome, faulty candidate(s)> pairs. Such a log may be 1) generated from a model of the board representing the relationship between components and tests, or 2) collected from previous diagnosis activities using a traditional approach based on complete syndromes.

When using previous diagnosis logs to train the decision tree, the used log must contain the required amount of information, for the learning process to be effective. Much research has been devoted to the extraction of an effective model of the system from a limited amount of historic data [10], [11]. In this work we do not address the problem of incomplete or insufficient information, focusing instead on the other alternative for building the decision tree, starting from a model provided by the test/diagnosis engineer, as the one used by commercial tools [12], [13] typically adopted in industrial environments.

1) *System Model*: We adopt the *Components-Tests Matrix* (CTM) system model proposed in [6], such that the overall proposal can be effectively integrated in a consolidated industrial diagnosis environment. The CTM model represents a complex electronic board in terms of its components (the

		Tests				
		T_1	T_2	T_3	T_4	T_5
Components	C_1	0.9	—	0.9	—	—
	C_2	0.5	0.9	—	0.1	0.1
	C_3	—	0.1	0.1	0.9	0.9

Fig. 2. Sample CTM used as a running example

rows of the matrix) and the tests that have been designed to exercise them (the columns of the matrix). Components consist of both ICs and passive components, and are of interest for the test engineer and for which at least one test exists. Each $ctm_{i,j}$ entry represents the probability that test T_j fails when component C_i is faulty. It is worth noting that, since it is very difficult to determine a quantitative estimation of this probability, a discrete, qualitative scale is used. More precisely, test engineers are required to specify how likely is that test T_j fails when component C_i is faulty, using values in the set $\{\text{High, Medium, Low, None}\}$. Then, in order for automatic tools to handle the CTM model of the board, the qualitative values $\{\text{High, Medium, Low, None}\}$ are translated into quantitative ones, using the scale $\{0.9, 0.5, 0.1, 0\}$. The limited impact of the chosen set of quantitative values on the behaviour of the diagnosis engines has been demonstrated ([8]). The CTM shown in Fig. 2 is adopted as running example throughout the paper.

2) *Diagnosis Log Generation*: The features of the problems are the available tests, the values of the features are the outcomes of the tests, i.e., either PASS or FAIL, and the class labels are the faulty candidate components. Thus, tests are associated with the internal nodes of the DT, test outcomes with branches and faulty candidate components with leaves. We refer to the set of faulty candidate components associated with leaf L_j as $FCCS_j$.

Provided a CTM model of the board under analysis is available, extracting a log from the CTM allows to keep track of all the possible syndromes. It is worth noting that, given a CTM model, not all syndromes are equally likely to occur. For instance, according to our running example, when C_3 is faulty, syndrome PPPFF is more likely to occur than PFFPP (the two occurrence probabilities are 21.89% and 0.000033%, respectively). We build the training log starting from the CTM mode. We calculate the probability of occurrence p_s of each syndrome s as:

$$p_s = \sum_{i=1}^{n_c} \left(AFP_i \cdot \prod_{j=1}^{n_t} p_{i,j} \right) \quad (1)$$

where AFP_i is the a-priori failure probability of component C_i (note that for the sake of simplicity, but without loss of generality, we are assuming that components have all the same a-priori failure probability) and $p_{i,j}$ is defined as:

$$p_{i,j} = \begin{cases} 1 - ctm_{i,j} & \text{if } s[T_j] = \text{PASS} \\ ctm_{i,j} & \text{if } s[T_j] = \text{FAIL} \end{cases} \quad (2)$$

In other words, $p_{i,j}$ represents the probability of test T_j having outcome $s[T_j]$ when the faulty component is C_i . When building the log, we include each syndrome in the log a number of times proportional to the occurrence probability of the syndrome itself. By including all the possible syndromes in the log we ensure that our approach is able to manage both frequent and very rare syndromes without compromising the

Test Outcomes					Diagnosis
T_1	T_2	T_3	T_4	T_5	
P	P	F	P	P	$\{C_1, C_3\}$
F	P	P	P	P	$\{C_1, C_2\}$
F	P	F	P	P	$\{C_1\}$
P	P	P	P	F	$\{C_2, C_3\}$
P	F	P	F	F	$\{C_2, C_3\}$
			⋮		
F	P	P	P	F	$\{C_2\}$
F	F	P	P	P	$\{C_2\}$
P	P	F	F	F	$\{C_3\}$
P	F	F	F	F	$\{C_3\}$

Fig. 3. An excerpt of the log extracted from the example CTM

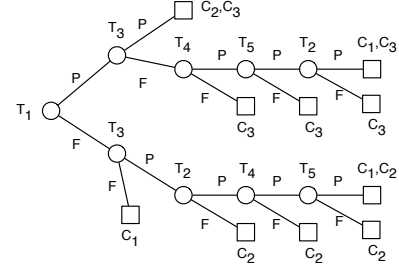


Fig. 4. The decision tree associated with the CTM shown in Figure 2.

diagnosis accuracy; on the other hand, as we will discuss in the next section, when dealing with frequent syndromes, the approach will be much more effective than when dealing with rare syndromes in terms of number of executed tests. Figure 3 shows an excerpt of the log extracted from the example CTM, which corresponds to the log collected from historic test data.

It is worth noting that some complete syndromes may point to more than one faulty candidate component. In this condition, no diagnosis engine can exactly identify the faulty component. Thus, the best result without introducing possible errors is to identify as faulty candidates all the components associated with the syndrome. This may happen in the scenario where the previous diagnosis logs are used, as well as when using the CTM model, since it is an issue related to the lack of isolation capability of the available tests.

3) *Decision Tree Extraction*: By starting from the log, the DT for the board under consideration can be extracted and trained and Figure 4 reports the DT associated with the running example CTM. The DT itself intrinsically encodes the test ordering policy and the stop condition, determined by the tree traversal from root to leaves, and the natural conclusion of the diagnosis process occurs when a leaf is reached.

A. Diagnosis Process

When performing the incremental process, the following steps are iteratively carried out:

- test selection: identification of the most useful test to be executed given the current partial available information (outcomes of the already executed tests),
- stop condition evaluation: analysis of the current partial information to determine whether to stop the procedure because one or more faulty candidate components can be identified, or to continue with other tests, and

faulty/not faulty components identification: based on the latest collected observation, analysis of the not yet classified components, that can now be identified as the root cause(s) of the system failure, or undoubtedly fault free.

Once the DT associated with a board has been built, when a failing device is found, the diagnosis process is executed, by exploiting the DT and traversing it. More in details, the first test to be executed is always the one associated with the root; based on the test outcome, a branch will be taken, reaching one of the two children. The process continues until a leaf is reached, and the associated components are identified as faulty candidates.

By referring to the DT in Figure 4, let us consider the case where the complete syndrome would be PPFPP, pointing to component C_3 . The incremental process behaves as follows. T_1 is the root of the DT and is initially executed and passes, leading to T_3 as the next test to be executed. Being $T_3=FAIL$, the next test to be executed is T_4 , with outcome FAIL and a leaf is reached, identifying C_3 as the faulty component.

As previously discussed, frequent syndromes appear in the training set many more times than rare ones, therefore the proposed approach behaves much better when dealing with the former ones, as the decision tree considers them *more reliable*. As an example, we consider the two previously mentioned syndromes: PPPFF and PFFPP. When dealing with PPPFF, which is a frequent syndrome, the proposed approach only requires the execution of T_1 and T_3 to identify C_2 and C_3 as the faulty candidate components and C_1 as a not faulty component. On the other hand, when dealing with PFFPP, which is a very rare syndrome, the approach requires to execute the whole suite of tests to identify C_3 as the faulty component and C_1 and C_2 as fault free. Actually the behaviour mimics that of an expert, focusing first on the most probable causes and only later resorting to the least frequent causes, when all previous observations have not produced any diagnosis.

For a more detailed discussion on the metrics that we adopted to evaluate the effectiveness and efficiency of the proposed approach in terms of accuracy and number of executed tests we refer the reader to Section V-A.

IV. THE PROPOSED EARLY STOP CONDITION

To speed up the diagnosis process we designed an *early stop condition*, to further reduce the number of tests required to identify the faulty candidate component(s) leveraging on the diagnosis accuracy. The early stop condition is based on a forward exploration of the tree, and verifies whether the maximum error (Max_e) that would be introduced by interrupting the process before reaching a leaf is smaller than a configurable threshold value Err_{th} .

To support this behavior, we introduce three sets of components associated with the DT: *Faulty Components Set (FCS)*, *Not Faulty Components Set (NFCS)* and *Unclassified Components Set (UCS)*. FCS contains the components identified as faulty, NFCS the components identified as fault-free, and UCS the component component for which no classification applies. Given a failing board, at the beginning of the diagnosis process, FCS and NFCS are empty while UCS contains all the components. During the traversal of the tree, when a test T_j

fails, all the components having $ctm_{ij} = 0$ are moved from UCS to NFCS; in fact, $ctm_{ij} = 0$ means when component C_i is faulty test T_j does not fail, consequently T_j failing implies C_i not faulty. This allows to have an updated forecast of the evolution of the diagnosis process by analyzing these sets, so to be able to decide whether to interrupt the process in favour of a reduced number of tests. Moreover, a set of faulty candidate components is associated with each internal node N_i of the DT: this set, referred to as *reachable faulty candidate components set (RFCCS_i)*, contains all the faulty candidate components associated with all the leaves belonging to the sub-tree having internal node N_i as root node.

In order to evaluate costs and benefits of an early interruption when at node N_i , the following maximum error estimation Max_e is introduced:

$$Max_e = 1 - \frac{\#FCCS_{min_i}}{\#RFCCS_i} \quad (3)$$

where $\#FCCS_{min}$ is the minimum cardinality among the ones of all the faulty candidate component sets of the leaves belonging to the sub-tree having N_i as root and $\#RFCCS_i$ is the cardinality of the reachable candidate component set of N_i . If the early stop condition is verified, all the components in the reachable faulty components set of internal node N_i are moved in FCS and the diagnosis procedure stops.

Let us recall the previous running example, with complete syndrome PPFPP and faulty component C_3 , and let us set $Err_{th} = 60\%$. After executing T_1 , with outcome PASS, the traversing of the tree reaches the node associated with T_3 : this node has $\#FCCS_{min} = 1$ and $\#RFCCS = 3$ (indeed, the possible diagnoses are $\{C_2, C_3\}$, $\{C_3\}$ and $\{C_1, C_3\}$); thus the maximum error introduced by the early stop condition at the current step of the execution is 66.66%, higher than the threshold, thus the process proceeds with test T_3 . $T_3=FAIL$ leads to node associated with T_4 . This node has $\#FCCS_{min} = 1$ and $\#RFCCS = 2$ (the possible diagnoses are $\{C_3\}$ and $\{C_1, C_3\}$), and $Max_e = 50\%$, which allows to stop the diagnosis process. C_1 and C_3 are identified as faulty candidates. The loss of accuracy with respect to the diagnosis procedure without early stop condition is 50%, the reduction in the number of executed tests is 33.33%.

V. EXPERIMENTAL ANALYSIS

A. Evaluation Metrics

We defined the following two metrics to evaluate the effectiveness of the proposed approach: i) the accuracy of the diagnosis, *Acc.* with respect to the outcome achieved by a diagnosis procedure based on the execution of all tests, and ii) the number of executed tests, $\#T$. As previously discussed, it is possible that, according to the CTM model of the board, some syndromes point to more than one faulty candidate. In this condition neither a *traditional* diagnosis procedure nor an incremental one could discriminate among the faulty candidates and identify the exact faulty component. Given this, the proposed approach can either identify the same faulty candidate(s) as a traditional approach (possibly executing fewer tests) or introduce inaccuracies by identifying either a subset of the faulty candidates (false negatives) or additional ones (false positives).

TABLE I. BOARDS CHARACTERISTICS.

ID	n_C	n_T	n_{LS}	n_{FS}	n_{SS}	n_{RS}
1	4	9	98	17	36	45
2	5	13	415	19	66	330
3	7	17	2383	26	132	2225
4	10	14	1090	27	143	920
5	10	18	1678	56	94	1528
6	15	25	1249	19	109	1121
7	19	29	1339	22	134	1183
8	22	32	953	25	183	745
9	25	40	3063	18	141	2904
10	32	55	4677	15	196	4466
Ind. 1	5	18	11788	14	72	11702
Ind. 2	14	24	12290	16	20	12254
Ind. 3	25	44	13261	17	39	13205

Furthermore, different syndromes have different occurrence probabilities, thus the two mentioned metrics $Acc.$ and $\#T$ are computed by weighting the contributions on the probability of occurrence of the syndromes; the impact of the behaviour of the approach for syndromes that rarely occur is less relevant than that of frequent syndromes. We calculated these two indices as:

$$Acc = \sum_{s=1}^{n_{LS}} (acc_s \times p_s) \quad (4)$$

$$\#T = \sum_{s=1}^{n_{LS}} (num_tests_s \times p_s) \quad (5)$$

where acc_s and num_tests_s are the accuracy and the number of executed tests, respectively, when analyzing syndrome s , and p_s is the occurrence probability of syndrome s calculated with Equation 1. The accuracy of the approach for a given syndrome s is calculated as:

$$acc_s = \frac{N_{Corr_s}}{N_{Tot_s} + N_{NotCorr_s}} \quad (6)$$

where N_{Corr_s} is the number of components correctly diagnosed, N_{Tot_s} is the total number of components associated with syndrome s and $N_{NotCorr_s}$ is the number of components wrongly diagnosed (either false positives or false negatives);

B. The Considered Boards

We evaluated the proposed approach on a set of 10 synthetic but realistic boards of various sizes and complexity, whose characteristics are reported in Table I, in terms of number of components (n_C), number of tests (n_T) and number of legal syndromes (n_{LS}) compatible with the CTM defining the board model. The last three rows report the details of three real-world industrial boards. Syndromes have been clustered according to their occurrence probability. The second part of Table I reports the number of *frequent* (column n_{FS}), *sporadic* (column n_{SS}) and *rare* (column n_{RS}) syndromes, having a probability of occurrence higher than 1%, between 1% and 0.1% and lower than 0.1%, respectively.

C. Results and Discussion

All the experiments have been carried out by considering one complete syndrome at a time, knowing the associated faulty component(s) identified by a process using the complete syndrome. Given such “hidden complete” syndrome (and associated “golden” diagnosis), we execute our methodology and perform the dynamically selected sequence of tests until the stop condition is verified. At that point we compare

the outcomes of the proposed method against the “golden” diagnosis and compute the accuracy. The process is repeated for all the legal syndromes of each considered CTM.

Table II reports the results of the experiments. More in details, columns two and three report the accuracy and the percentage of executed tests with respect to the whole test suite when using the basic stop condition (considered as the “baseline experiment”). The remaining columns report the reduction of accuracy and of the number of executed tests with respect to the baseline experiment when considering the early stop condition with $Err_{th} = 25\%$, 50% and 75% , respectively.

Do note that the basic execution of the proposed approach (without early stop condition) achieves a high fault coverage for most of the considered boards (92.85% on the average) using in most cases less than the 50% of the available tests (44.62% on average). If we look at the remaining results, it can be observed that, while $Err_{th} = 25\%$ does not bring any noticeable benefit, the execution with the early stop condition allows to significantly reduce the amount of executed tests when $Err_{th} = 50\%$ and $Err_{th} = 75\%$: -13.72% and -30.14% on average, respectively, with a maximum of -39.79% and -77.71%, respectively. Accuracy decrease is limited (-11.20% and -28.51% on average, respectively) and in most cases is smaller than the decrease in the number of executed tests.

The effectiveness of the proposed approach can be better highlighted by focusing on frequent syndromes, and Table III reports the detailed results of the previous experiments for these syndromes only. The baseline experiment achieves always a 100% accuracy (except for boards 11 and 12) with a reduced percentage of executed tests (42.17% on the average). Case $Err_{th} = 25\%$ has no significant effects, while when $Err_{th} = 50\%$ and $Err_{th} = 75\%$, the reduction of executed tests is large (-15.99% and -34.83% on average, respectively, with maximum values of -42.68% and -78.77%, respectively) while the decrease of the accuracy is limited (-6.88% and -21.89% on average, respectively) and in almost all cases (apart boards 5, 10 and 12) is always significantly smaller than the reduction of the number of executed tests. In some cases there is even a significant reduction of the number of tests (-5.88% for board 6 and -13.43% for the industrial board 3) with no decrease in accuracy. As a result, when the user is interested in an aggressive reduction of the number of tests, the loss in accuracy is never as harsh as the benefit in terms of test time/effort. This is the tunable element the user can exploit while fitting her/his requirements and constraints.

Experiments have been carried out on an Intel i7-3770 3.4GHz microprocessor equipped with 8GB of RAM; the time required to carry out the analysis varies from about 1 second for the smallest synthetic boards to half a minute for the largest ones, and about 10 seconds for the industrial boards. Therefore, we believe that the proposed diagnosis approach could easily be integrated in an industrial test process.

For the sake of space, in this paper we focused on the assessment of the effectiveness and efficiency of the proposed decision tree-based diagnosis engine and of its early stop condition. We refer the reader to [8] for a thorough comparison among several machine learning-based diagnosis engines in terms of accuracy as well as scalability and robustness against errors in the CTM model.

TABLE II. RESULTS ACHIEVED BY THE PROPOSED DIAGNOSIS APPROACH.

boardID	$Err_{th} = 0\%$		$Err_{th} = 25\%$		$Err_{th} = 50\%$		$Err_{th} = 75\%$	
	Acc. (%)	T (%)	Δ Acc. (%)	Δ T (%)	Δ Acc. (%)	Δ T (%)	Δ Acc.(%)	Δ T (%)
1	100.00%	53.33%	-0.00%	-0.00%	-32.07%	-39.79%	-50.02%	-77.71%
2	100.00%	36.46%	-0.00%	-0.00%	-16.30%	-38.82%	-16.47%	-41.98%
3	100.00%	46.30%	-0.04%	-0.30%	-6.46%	-27.41%	-20.25%	-35.38%
4	100.00%	66.57%	-0.10%	-0.40%	-7.22%	-15.77%	-39.72%	-46.03%
5	99.99%	45.16%	-0.00%	-0.00%	-21.30%	-9.59%	-51.89%	-28.17%
6	100.00%	42.40%	-0.00%	-0.00%	-2.03%	-7.08%	-22.61%	-27.76%
7	99.50%	45.72%	-0.06%	-0.60%	-6.80%	-7.54%	-22.32%	-25.49%
8	99.19%	41.49%	-0.00%	-0.00%	-4.00%	-7.08%	-31.11%	-25.17%
9	96.15%	38.95%	-0.07%	-0.10%	-8.39%	-7.31%	-22.95%	-26.32%
10	84.27%	31.64%	-0.00%	-0.00%	-10.40%	-5.52%	-39.50%	-18.80%
11	64.85%	26.83%	-0.00%	-0.00%	-8.82%	-2.91%	-21.31%	-9.60%
12	49.76%	19.75%	-0.00%	-0.00%	-13.44%	-2.67%	-39.93%	-8.19%
Ind. 1	100.00%	41.50%	-0.00%	-0.00%	-20.13%	-21.29%	-23.17%	-49.53%
Ind. 2	100.00%	69.75%	-0.00%	-0.00%	-5.49%	-9.14%	-14.07%	-22.46%
Ind. 3	98.99%	63.41%	-0.00%	-0.00%	-5.17%	-3.91%	-12.31%	-9.53%
avg	92.85%	44.62%	-0.02%	-0.09%	-11.20%	-13.72%	-28.51%	-30.14%

TABLE III. DETAILED ANALYSIS OF THE RESULTS ACHIEVED BY THE PROPOSED DIAGNOSIS APPROACH ON THE FREQUENT SYNDROMES.

boardID	$Err_{th} = 0\%$		$Err_{th} = 25\%$		$Err_{th} = 50\%$		$Err_{th} = 75\%$	
	Acc. (%)	T (%)	Δ Acc. (%)	Δ T (%)	Δ Acc. (%)	Δ T (%)	Δ Acc.(%)	Δ T (%)
1	100.00%	52.33%	-0.00%	-0.00%	-29.22%	-42.68%	-47.31%	-78.77%
2	100.00%	33.92%	-0.00%	-0.00%	-12.66%	-42.63%	-12.66%	-46.94%
3	100.00%	43.62%	-0.00%	-0.00%	-2.07%	-28.75%	-20.09%	-35.10%
4	100.00%	71.64%	-0.00%	-0.30%	-3.11%	-17.85%	-29.34%	-46.86%
5	100.00%	39.67%	-0.00%	-0.00%	-21.43%	-6.02%	-59.53%	-14.00%
6	100.00%	38.76%	-0.00%	-0.00%	-0.00%	-5.88%	-17.14%	-33.02%
7	100.00%	40.17%	-0.74%	-0.69%	-4.75%	-10.82%	-13.55%	-35.11%
8	100.00%	42.69%	-0.00%	-0.00%	-1.87%	-8.64%	-18.21%	-28.55%
9	100.00%	38.32%	-0.00%	-0.33%	-4.24%	-5.35%	-8.55%	-33.72%
10	100.00%	31.78%	-0.00%	-0.00%	-5.22%	-5.32%	-27.97%	-21.11%
11	93.93%	23.09%	-0.00%	-0.00%	-2.15%	-5.53%	-7.04%	-15.94%
12	50.00%	22.72%	-0.00%	-0.00%	-0.00%	-0.00%	-25.00%	-4.00%
Ind. 1	100.00%	52.56%	-0.00%	-0.00%	-12.21%	-22.09%	-19.76%	-62.05%
Ind. 2	100.00%	60.54%	-0.00%	-0.00%	-4.23%	-24.91%	-11.28%	-26.43%
Ind. 3	100.00%	40.79%	-0.00%	-0.00%	-0.00%	-13.43%	-10.96%	-40.89%
avg	96.26%	42.17%	-0.05%	-0.09%	-6.88%	-15.99%	-21.89%	-34.83%

VI. CONCLUSIONS AND FUTURE WORK

We have presented a new approach for adaptive incremental functional diagnosis at the board-level based on the decision trees. The approach has demonstrated to be effective in achieving a high diagnosis accuracy and in reducing the number of tests required to be executed with respect of the whole test suite. Moreover, we have presented a configurable early stop condition for the proposed approach that allows to further reduce the number of executed test without excessively reducing the accuracy of the diagnosis. By configuring the stop condition, the user is empowered to tune the reduction of number of executed tests and the decrease of the accuracy.

As future work we plan to consider and compare different algorithms for the generation of the decision tree in order to further improve the efficiency of the proposed method.

ACKNOWLEDGEMENTS

This work is partially supported by the Cisco University Research Program Fund Gift #2014-130689 (3696), an advised fund of Silicon Valley Community Foundation.

REFERENCES

[1] P. Maxwell, I. Hartanto, and L. Bentz, "Comparing functional and structural tests," in *Proc. Int. Test Conf.*, 2000, pp. 400–407.

[2] F. Ye, Z. Zhang, K. Chakrabarty, and X. Gu, "Adaptive board-level functional fault diagnosis using decision trees," in *Proc. Asian Test Symp.*, 2012, pp. 202–207.

[3] —, "Board-level functional fault diagnosis using learning based on incremental support-vector machines," in *Proc. Asian Test Symp.*, 2012, pp. 208–213.

[4] —, "Board-level functional fault diagnosis using artificial neural networks, support-vector machines, and weighted-majority voting," *IEEE Trans. Computer-Aided Design of Integrated Circuits and Systems*, vol. 32, no. 5, pp. 723–736, 2013.

[5] Z. Zhang, Z. Wang, X. Gu, and K. Chakrabarty, "Board-level fault diagnosis using bayesian inference," in *Proc. VLSI Test Symp.*, 2010, pp. 244–249.

[6] L. Amati, C. Bolchini, L. Frigerio, F. Salice, B. Eklow, A. Suvatne, E. Brambilla, F. Franzoso, and M. Martin, "An incremental approach to functional diagnosis," in *Proc. Int. Symp. Defect and Fault Tolerance in VLSI Systems*, 2009, pp. 392–400.

[7] C. Bolchini, E. Quintarelli, F. Salice, and P. Garza, "An expert cad flow for incremental functional diagnosis of complex electronic boards," *IEEE Trans. Computer-Aided Design of Integrated Circuits and Systems*, vol. 34, no. 5, pp. 835–848, Jan. 2015.

[8] C. Bolchini and L. Cassano, "Machine learning-based techniques for incremental functional diagnosis: A comparative analysis," in *Proc. Int. Symp. Defect and Fault Tolerance in VLSI and Nanotechnology Systems*, 2014, pp. 246–251.

[9] S. B. Kotsiantis, "Supervised machine learning: a review of classification techniques," *Informatica*, vol. 31, no. 3, 2007.

[10] H. Wang, O. Poku, X. Yu, S. Liu, I. Komara, and R. D. Blanton, "Test-data volume optimization for diagnosis," in *Proc. Design Automation Conf.*, 2012, pp. 567–572.

[11] F. Ye, Z. Zhang, K. Chakrabarty, and X. Gu, "Information-theoretic syndrome and root-cause analysis for guiding board-level fault diagnosis," in *Proc. IEEE European Test Symposium*, 2013, pp. 1–6.

[12] *Fault Detective, 4.0*, Agilent Technologies. [Online]. Available: www.agilent.com/find/fd

[13] L. Bardford, V. Kanevsky, and L. Kamas, "Bayesian fault diagnosis in large-scale measurement systems," in *Proc. Instrumentation and Measurement Technology Conf.*, 2004, pp. 1234–1239.