electronics

MDPI

Article

Identifying the Optimal Subsets of Test Items through Adaptive Test for Cost Reduction of ICs

Huaguo Liang, Jinlei Wan *, Tai Song and Wangchao Hou

School of Electronic Science & Applied Physics, Hefei University of Technology, Hefei 230009, China; huagulg@hfut.edu.cn (H.L.); baikediguo@hfut.edu.cn (T.S.); 2018111142@mail.hfut.edu.cn (W.H.)
* Correspondence: 2019111134@mail.hfut.edu.cn

Abstract: With the growing complexity of integrated circuits (ICs), more and more test items are required in testing. However, the large number of invalid items (which narrowly pass the test) continues to increase the test time and, consequently, test costs. Aiming to address the problems of long test time and reduced test item efficiency, this paper presents a method which combines a fast correlation-based filter (FCBF) and a weighted naive Bayesian model which can identify the most effective items and make accurate quality predictions. Experimental results demonstrate that the proposed method reduces test time by around 2.59% and leads to fewer test escapes compared with the recently adopted test methods. The study shows that the proposed method can effectively reduce the test cost without jeopardizing test quality excessively.

Keywords: integrated circuit test; cost reduction; adaptive test; naive Bayesian model; fast correlation-based filter

check for updates

Citation: Liang, H.; Wan, J.; Song, T.; Hou, W. Identifying the Optimal Subsets of Test Items through Adaptive Test for Cost Reduction of ICs. *Electronics* **2021**, *10*, 680. http://doi.org/10.3390/electronics10060680

Academic Editor: Kiat Seng Yeo

Received: 18 February 2021
Accepted: 11 March 2021
Published: 14 March 2021

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Copyright: © 2021 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

As the scaling of ICs (integrated circuits) continues, more advanced and complex structures demand comprehensive tests [1]. This results in a significant increase in test costs to ensure high test quality (fewer test escapes). According to the International Technology Roadmap for Semiconductors (ITRS), test cost is an important part of the overall manufacturing cost [2]. Therefore, reducing cost without jeopardizing quality represents a pressing concern.

To achieve an acceptable defect level for the manufactured chips, a large number of test items are applied to measure current, voltage, and other parametric values [3]. A chip is considered good only if it passes all items. By means of a large number of test items, defective chips are weeded out. As defects become more complicated, more test items are needed, thereby incurring increasing test costs [4].

The higher the number of test items, the higher the number of irrelevant or redundant ones, thereby reducing the effectiveness of the test items [5]. Chips coming from the same fabs, the same batch or the same flow may share similar defects [6]. Therefore, it is unnecessary to use the full items for testing; the most appropriate method is to select a subset of specification parameters to measure and achieve an acceptable defect level [7].

An adaptive test framework is now being advocated to replace the traditional testing approach, and it provides a new strategy to reduce test costs [8]. In the adaptive test, each circuit can be tested with a unique process, which may involve adjusting the test items; that is, given a test suite, certain ineffective items are eliminated so as to reduce the testing time [9,10].

Great effort has been expended on adaptive test methods that eliminate redundant items to reduce test time. To identify the optimal subset, Xue et al. [11] analyzed and selected each test one by one in a streaming fashion to save both time and memory. M. Grady et al. [12] applied a greed algorithm to obtain an optimal subset from the observation of samples. Wang et al. [13] built a cost model to select patterns dynamically to optimize the test cost. Agrawal et al. [14] proposed a similar cost model based on a genetic algorithm to minimize the test set. The above approaches focused on selecting the optimal test set, but they did not consider the correction between tests, which limits the reduction in cost. To address this issue, Ahmadi et al. [9] proposed a method to dynamically select whether to subject a wafer to a complete or a reduced test set based on an e-test signature. Xin Li et al. [15] proposed an iterative test selection method with Bayesian index for test cost reduction. The above approaches attempt to make a prediction considering the trade-off between test time and test quality. However, they only consider a linear correlation between tests and they lack preprocessing, and the accuracy of the prediction model still requires improvement. Chakrabarty et al. [4] proposed a fine-grained adaptive test method based on machine learning to make quality predictions. Shambhu et al. [16] adaptively assigned the cores in parallel to the test access mechanisms with variable widths. However, it is difficult to build a model for the above approaches and some implementations even need additional hardware overhead. Therefore, they cannot address the problem of broader applicability and generality.

As demonstrated, most of the aforementioned methods fail to sufficiently consider the correction and characteristics of the items when trying to minimize test cost. They do not adequately consider the test quality and there has been relatively little research on identifying effective test item methods by means of mathematical models without jeopardizing test quality excessively.

In this work, we proposed an adaptive test method which combines a fast correlation-based filter (FCBF) and the weighted naive Bayesian model to reduce test costs. Figure 1 shows the entire adaptive test flow. First, a complete test was applied to the randomly selected sample chips. Based on the test outcomes, the proposed method obtains an optimal subset and prediction model to achieve a significant test time reduction without excessively sacrificing test escape.

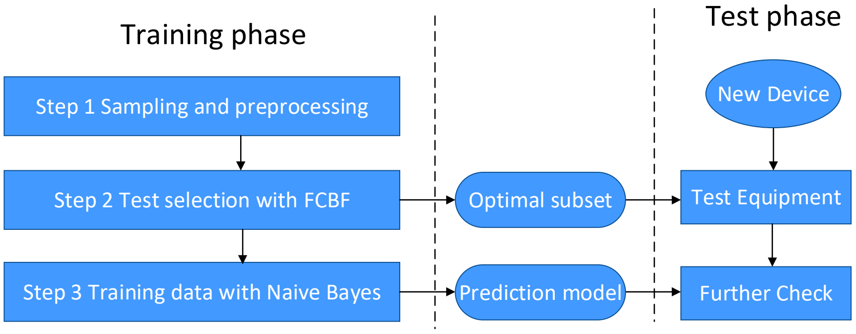

Training phase
Test phase
Step 1 Sampling and preprocessing
Step 2 Test selection with FCBF
Step 3 Training data with Naive Bayes
Optimal subset
Prediction model
New Device
Test Equipment
Further Check

Figure 1. The proposed adaptive test flow.

The main contributions of the paper are as follows:

1. Test data of chips are preprocessed based on the characteristics of items to improve prediction accuracy;
2. FCBF identifies effective test items so as to reduce test cost, and a weighted naive Bayes model is trained to predict the quality of chips based on the outcomes of selected ones;
3. The quality of each chip is used to select an appropriate test set for chips with comparable quality.

The rest of the paper is structured as follows. In Section 2, the background and strategy are shown. In Section 3, the proposed method is presented in detail. Section 4 shows experimental outcomes, and the conclusions are presented in Section 5.

2. Background

2.1. Adaptive Test for Parametric Test

In general, each chip is subjected to two types of tests: parametric tests and functional tests. Parametric tests include DC (Direct Current) and AC (Alternating Current) parametric tests to measure different values [17]. Upper and lower bounds of each test item are provided with test specification. A chip passes a test item if the outcome is within the acceptable range. In the traditional tests, these parametric results are only used to determine whether the chips pass or fail, and the analysis of them is ignored [18].

In an adaptive test, the test content, test order and pass/fail limits are not fixed, as in conventional tests, but depend on the test results of the currently or historically tested data. In the extreme, each wafer or die can be tested using a unique test process [19]. In this way, significant test time savings and test quality improvements can be achieved [20].

2.2. Naive Bayesian Model

The naive Bayesian model comes from classical mathematical theory and has stable classification efficiency. It works based on Bayesian rules and probability theorems [21]. Let $(f_1, \cdots f_n)$ represent n features and c represent the class label. In order to predict the class label, the probability that the target belongs to class c is computed according to the features. The probability can be obtained as Equation (1):

$$P(c|(f_1, \cdots f_n)) = \frac{p(c)\prod_{i=1}^{n} p(f_i|c)}{\prod_{i=1}^{n} p(f_i)} \tag{1}$$

Through the naive Bayesian model, the probability that a chip will "pass" can be obtained. It will be used as the quality prediction result in the following. Chips will be dealt with according to the prediction.

This model is adopted here because it is not overly sensitive to missing data and the algorithm is relatively simple, which means that it is easier to implement during the test. However, the performance of the naive Bayesian model is always limited by the mathematical assumption of the model that each feature is independent. As mentioned before, it is common that items share some corrections. Test selection is necessary here because it incurs a test cost reduction and improves the accuracy of prediction [22]. Accurate prediction allows us to remove more items and helps to identify potential defective chips. The FCBF which is proposed in [23] is also incorporated in order to solve the problem.

3. Proposed Testing Method

The entire adaptive test flow includes a training phase and test phase. The training consists of three stages: (1) data sampling and preprocessing; (2) application of test selection algorithm to obtain optimal subsets; (3) training of the naive Bayesian model to obtain the prediction model.

For the test chips, the test phase is shown in Figure 2. A chip is tested with the selected items at first. Once it passes all selected ones, the prediction model is used to make a quality prediction. The goal of the prediction is to identify suspicious items that can pass all selected tests but behave differently. In order to guarantee the quality of the test, a higher threshold is set in order to enable stringent judgments. Once a chip is judged as a suspicious one, the items removed by the selection algorithm will be applied to conduct a further check.

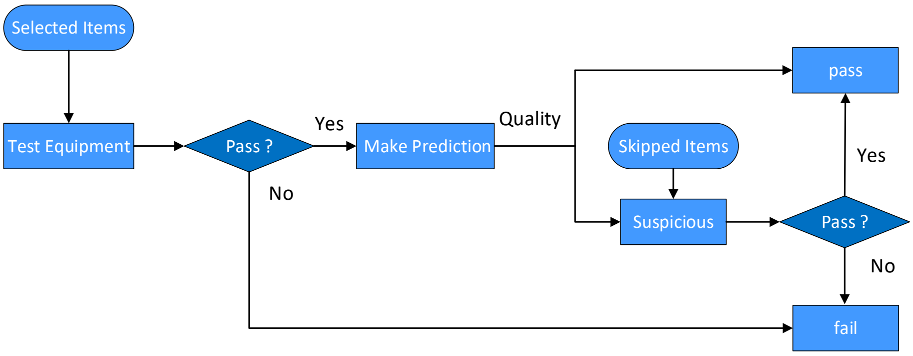


Figure 2. The test flow for the test chips.

3.1. Data Preprocessing

In order to realize the selection and prediction flow, the samples need to be processed before learning. There are two existing problems here: (1) missing data; (2) imbalanced original data. These factors mean that even adding or deleting a small amount may lead to significant changes in classification.

Test data may be missing because a test item is performed only if all previous items pass the specifications according to the stop-at-first-fail strategy of Automatic Test Equipment (ATE). In addition, results may be lost if the database is corrupted. Missing data will be handled as follows: items will be removed from the training set if more than 10% are missing. Otherwise, the missing parts are filled with the median of the available data associated with the nearest 100 results of the same item. With the nearest results, the filled value will be closer to the originally expected value because of the process shift.

The imbalanced data exist because, in general, most chips will pass all test items, and analytical methods cannot perform as well as hoped in the face of imbalanced data. An oversample of the failing data is taken in order to change this. Adaptive synthetic sampling (ADASYN) is adopted here to obtain more failing data. ADASYN is one of the SMOTE (Synthetic Minority Oversampling Technique) variants. SMOTE generates new examples along the line between the failing ones and their selected nearest points, and the number of new examples generated by each minority class is the same. The ADASYN algorithm introduces a criterion to automatically determine the number of synthetic samples that need to be generated for each minority data example [24]. New points will be added to the training set to obtain balanced data.

After the processing, the result of pass/fail is collected as the target class label c, and the result of item i will be trained as feature f_1 in the following learning process.

3.2. Test Selection Algorithm

The goal of selection is to obtain a subset of items to reduce the test cost while maintaining the test quality and prediction accuracy. A test item is selected if it is relevant to the class but not redundant for any other relevant items [25]. It is a multivariate feature selection method which uses symmetrical uncertainty (SU) to calculate dependencies to select the optimal subset. The algorithm consists of two steps. It calculates corrections between the target class and each feature to remove irrelevant features first. Then, an iteration is taken to reduce redundancy. During the selection, the cost of each item is also taken into account to obtain a better cost reduction.

The symmetrical uncertainty (SU) is used to estimate the correction between two variables. The SU is a normalized information theoretic measure and uses entropy and conditional entropy values to calculate dependencies. The entropy and conditional entropy need to be calculated first.

If X is a random variable and $P(X)$ is the probability of x, the entropy which represents the uncertainty of X is:

$$H(X) = -\sum_i P(x_i)\log_2 P(x_i) \tag{2}$$

Given another random variable Y, the conditional entropy $H(X|Y)$ measures the uncertainty of the value of X given the value Y and is defined as:

$$H(X|Y) = -\sum_{j} P(y_j) \sum_{i} P(x_i|y_i) \log_2 P(x_i y_j) \tag{3}$$

Then, SU can be calculated:

$$SU(X,Y) = 2\left[\frac{H(X) - H(X|Y)}{H(X) + H(Y)}\right] \tag{4}$$

where $H(X) - H(X|Y)$ indicates the information gain of X for Y, which represents the reduction in the uncertainty of X while Y is determined. An SU value of 1 indicates that the two variables can totally predict each other and a value of 0 indicates that the two variables are totally independent. The SU values are symmetric for both features.

First, the $SU(f_i, c)$ is calculated between each feature and class. The effectiveness of each feature is obtained, and a threshold δ is set in order to filter out inefficient ones. The time spent on each item is significantly different, and it is necessary to take this into account. The time spent on f_i is normalized to t_i. Using SU, the relevance between feature f_i and class c can be quantitatively compared. Combining this with t_i, the effective features can be screened out.

Then, the algorithm will calculate the minimum redundancy among the features to make the subset smaller. The features are reordered according to $SU(f_i, c)/t_i$. Starting from the first feature f_1, for each following feature, its correction is compared with the f_1 with its correction with class c. If $SU(f_i, f_1) \geq SU(f_i, c)$, the feature is redundant and can be removed. After f_1 comparison ends, f_2 continues. The iteration continues until the last one and the subset of retained features is decided.

3.3. Naive Bayesian Quality Prediction

After the selection, test cost is reduced and the correction of items is decreased so that the naive Bayesian model can perform better. It is a common assumption that the parametric test data obey Gaussian distribution. Thus, the $p(f_i|c)$ used in Equation (1) is rewritten as follows to replace the origin statistical value:

$$p(f_i|c) = \frac{1}{\sqrt{2\pi\sigma_{c,i}^2}} exp^{(-\frac{(f_i - \mu_{c,i})^2}{2\sigma_{c,i}^2})} \tag{5}$$

Although all items passed the screening, their ability to detect potential defects was not consistent and it was necessary to calculate their weights. In order to evaluate the capability of detecting potential failed chips, the C_{PK} is calculated as the metric. It is defined as in Equation (6):

$$C_{PK} = \min\left\{\frac{USL - \mu}{3\sigma}, \frac{\mu - LSL}{3\sigma}\right\} \tag{6}$$

where USL and LSL denote the upper and lower specification limits of each item. The lower C_{PK} is, the higher probability of detecting potential failed chips [4]. Normalizing $1/C_{PK,i}$ to $w(i)$ as the weight, Equation (1) becomes Equation (7):

$$P(c|(f_1, \cdots f_n)) = \frac{p(c) \prod_{i=1}^{n} p(f_i|c) w(i)}{\prod_{i=1}^{n} p(f_i)} \tag{7}$$

In order to make a string judgement, the threshold is adjusted. If the quality of a chip is lower than the threshold, it is suspicious. Through quality prediction, suspicious chips are identified and test quality is guaranteed.

4. Experimental Results

In the experiment, we employ the actual production data of a specific IC to evaluate the proposed method. It contains five wafers, each of which contains approximately 35,000 dies. In order to check the accuracy of the prediction, we consider a bad situation and choose the data coming from the batch where the faulty chips constitute around 12–14%. The standard test sets consist of 49 test items, where the connection test, DC test and power management unit test are covered. In wafer 1, 30% dies are collected as the training data. Then, the rest of the dies are tested with the selected subsets to validate the effectiveness, and the other wafers are used to check whether the learning result is stable. The experiment adopted Python 3.7 and skfeature, scikit-learn library. The random seed selection is 0.

After the preprocessing, the selection algorithm is used to obtain the subset items. $SU(f_i, c)$ is calculated first. Figure 3 shows the result of $SU(f_i, c)$. The threshold of $SU(f_i, c)$ is set as 0.18, and finally, 23 items are kept.

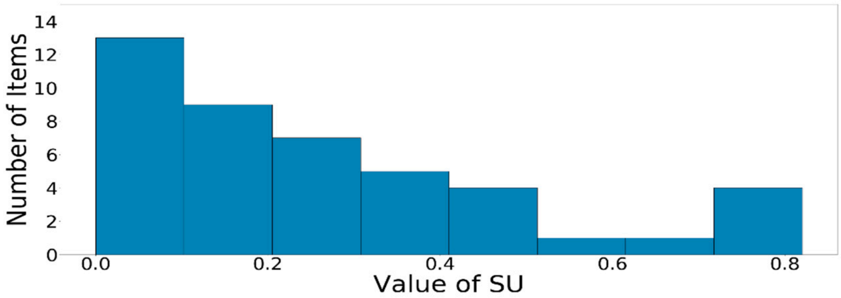


Figure 3. The distribution of $SU(f_i, c)$.

4.1. Performance of Quality Prediction

To make a comparison with other methods, threshold values need to be compared and decided first. A range of 0.6 to 1.0 was selected with steps of 0.05 to calculate the test escape (when faulty chips are undetected) and test time reduction (TTR).

The results of the different threshold values are shown in Figure 4. The turning point is between 0.75 and 0.8, and we choose 0.75 here because it achieved better TTR and nearly the same test escape compared with 0.8.

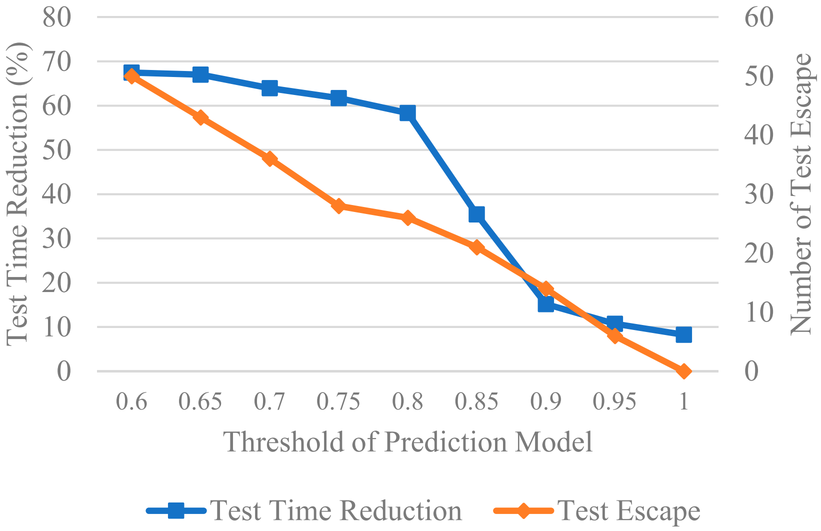


Figure 4. The results of different threshold values.

To evaluate the performance of quality prediction, we examine the percentage of failed chips in the two groups. Table 1 shows the comparison with various percentages of chips that are randomly selected as sample chips in threshold 0.75. All the experiments are repeated five times by randomly selecting a different set of sample chips each time; the results shown in the table are the average values over these five experiments.

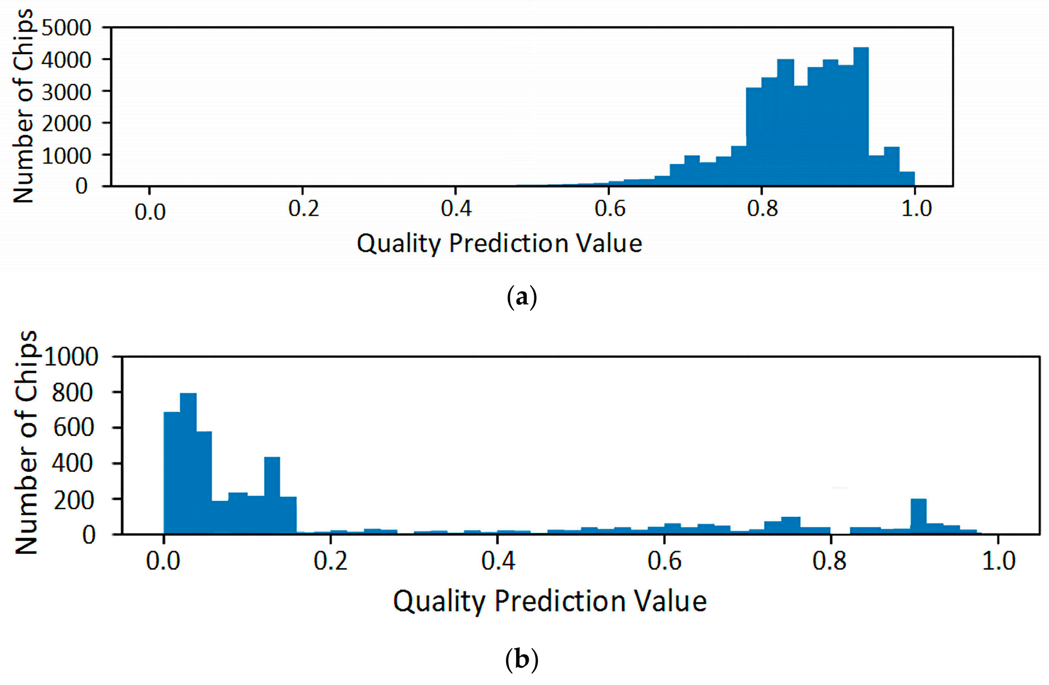
Table 1. Performance of quality prediction.

Percentage of Sample Chips	5%	10%	20%	30%	40%
Percentage of failed chips in "pass"	1.41%	1.26%	0.83%	0.26%	0.21%
Percentage of failed chips in "suspicious"	16.56%	18.43%	21.96%	23.75%	24.95%

The result improves with the increasing percentage of sample chips. This also indicates that threshold is a valuable metric that helps us in partitioning the chips into two groups, and we will apply it to the entire flow in future analyses.

We choose 30% as the sample rate in the following experiment and derive the distribution of quality prediction to show the relationship between quality and the pass/fail status of chips.

As shown in Figure 5, the distribution shows that the failed and passed chips can be distinguished well through the prediction result. We also choose one test item to test 10,000 chips to analyze the relationship, and the result is shown in Figure 6. The values of items are all within the original limits. If the prediction is made through just one item, the upper and the lower bounds' limits will be the line in the figure where the misclassification rate is approximately 14.4%.



(**a**)

(**b**)

Figure 5. Relationship between predicted quality and the pass/fail status of chips. (**a**) Passed chips; (**b**) failed chips.

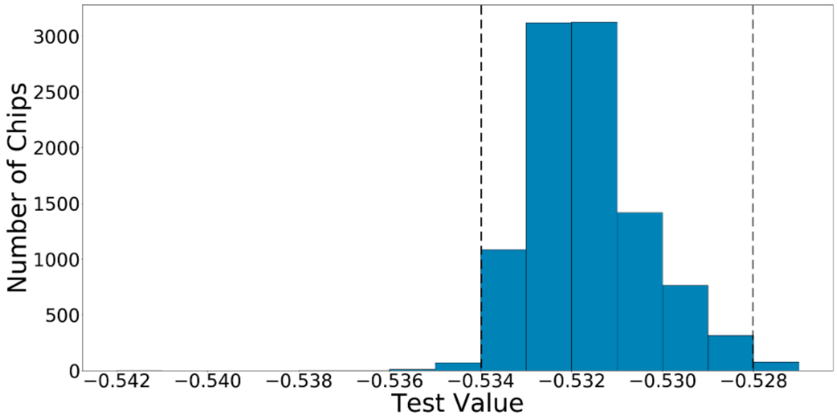


Figure 6. The results obtained using one test item.

Selecting those misclassified chips in Figure 5 for research, we find that they are mainly caused by some random defects. For these, the performance of the classification algorithm is limited unless a higher computational cost is paid. The outlier detection algorithm performs better in this regard [1], and better results may be obtained through weighted combination.

4.2. Comparison with Other Adaptive Test Methods

In order to evaluate the reduction in test cost and the effect on the test quality of the proposed method, the proposed method is compared with other adaptive methods here. The test result of wafer 1 is used for the comparison, and the traditional method is set as the baseline to measure the test escapes and TTR. Table 1 shows the TTR and test escapes of the proposed method and the other methods.

From Table 2, it is clear that the proposed method achieved a time saving of 61.72% but led to 28 test escapes compared with the traditional method, and these results are better than those of the other methods. The average items of [24] are close to the proposed one. However, the lack of consideration of test time during selection caused a significant difference in TTR. In [18], a similar TTR value with the proposed method is shown; nevertheless, there are more test escapes because it ignores the further analysis after selection.

Table 2. Comparison with other methods.

	Average Time (ms)	Average Items	TTR	Test Escapes
Traditional Method	1385	49	0.00%	0
Proposed Method	530.17	24	61.72%	14
Ref [11]	620.22	32	55.22%	28
Ref [19]	746.38	27	46.11%	15
Ref [15]	566.05	26	59.13%	26

Although the proposed method achieved the fewest escapes, this value still appears to be slightly higher than normal in terms of DPPM (Defect part per million). A higher rate of faulty chips brings more random defects, which makes the prediction more difficult, and we can achieve more accurate prediction in normal situations.

4.3. Further Comparison of Test Quality

To further compare the test quality, different proportions of items are used. The resampling was repeated five times to obtain the average value of the test escape rate.

Figure 7 shows the test escape rate while proportion ranged from 0 to 100%. The result clearly shows that a larger number of test items results in fewer test escapes. The proposed method achieved the smallest test escape rate compared with the other methods before the proportion reached 60%. After this, all methods achieved a low escape rate. The adaptive flow presented in [24] achieved similar performance in terms of quality as the proposed method. However, it sacrificed much more time than the other methods.

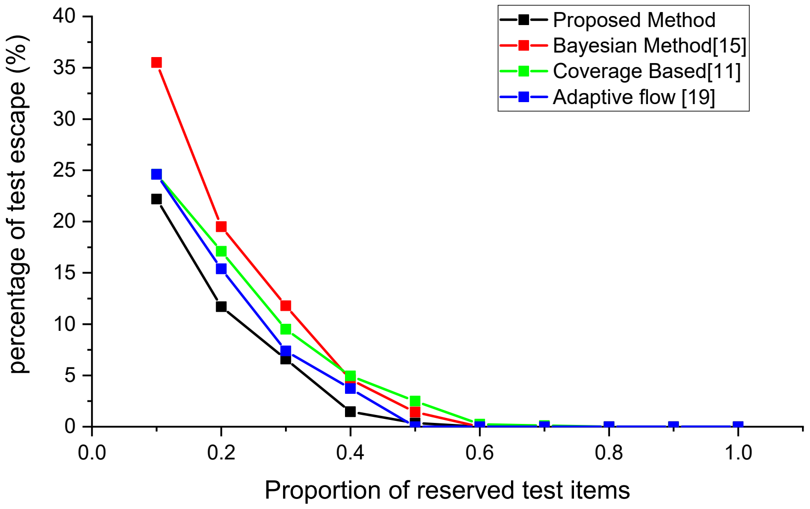


Figure 7. Test escape comparison.

The goal of the adaptive test is to identify the trade-off between test time and test escape according to the requirements. Figure 8 shows the test time needed compared with traditional methods and the test escape rate based on different proportions of test items. Because of the machine learning, the test escape can remain low even when a small number of tests are applied. In the experiment, around 40% items were retained where test escape rate was limited and test time was reduced.

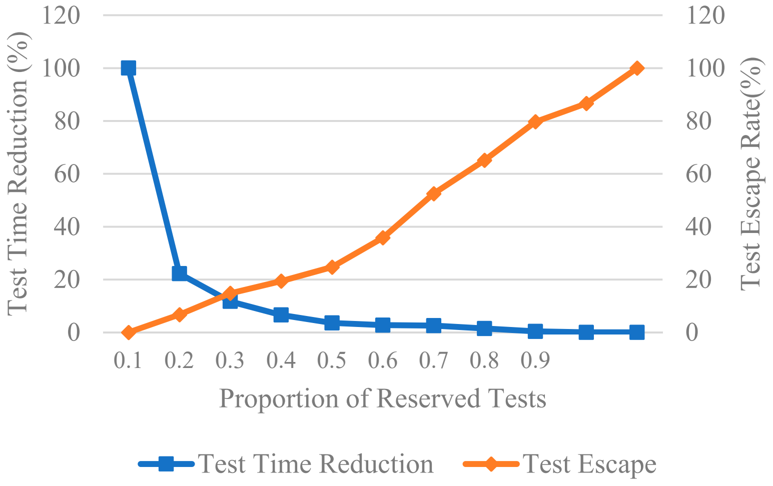


Figure 8. The result in different reserved proportions.

4.4. Comparison of Different Wafers

To check whether the effect of the proposed method can be kept constant, we apply the test flow to the rest of the wafers. Because of the process shift, the effectiveness of selected items and the accuracy of the prediction model may decrease. The comparison results when the proportion of test was around 50% are listed in Table 3.

Table 3. Results when using different wafers.

		Wafer 1	**Wafer 2**	**Wafer 3**	**Wafer 4**	**Wafer 5**
Proposed Method	TTR	61.72%	59.30%	62.52%	62.31%	58.10%
	Test Escape Rate	0.27%	0.56%	0.49%	0.72%	0.84%
Ref 2015 [11]	TTR	57.22%	55.72%	59.34%	58.66%	51.97%
	Test Escape Rate	0.60%	1.16%	1.47%	2.36%	2.98%
Ref 2018 [19]	TTR	46.11%	38.40%	27.51%	19.82%	17.39%
	Test Escape Rate	0.29%	0.60%	0.44%	0.38%	0.29%
Ref 2019 [15]	TTR	59.13%	58.43%	58.97%	58.30%	57.28%
	Test Escape Rate	0.51%	1.05%	1.11%	1.61%	1.99%

Table 3 shows the results of TTR and test escape rate in five different wafers with the proposed method and the other adaptive test methods. Because wafers are tested with the same items, the TTR demonstrates no great change, and the proposed method performs best. The TTR of [24] decreases clearly because it keeps adding items to its selected subset to ensure the test quality.

We can also see that the test escape rate in different wafers appears to show an upward trend, except in [24]. It achieved high test quality by using much more time. Compared with the other two methods, the proposed one yielded the fewest test escapes. The results show that the proposed method achieves the best compromise between test escape and test time.

Overall, the results of the five wafers show a downward trend because the prediction model cannot follow the changing process. If the parameters of the model can be updated promptly, the accuracy can be improved significantly. However, this method is limited by the online computation overhead. The research in this part focuses on how to design the pipelined test methodology [6,26] of ATE and develop efficient incremental learning methods [27,28]. This is an issue worth studying in future work.

5. Conclusions

The work of the paper focuses on the test cost reduction that can be achieved through identifying the most effective test items and making quality predictions. It only applies the test items with higher effectiveness and the comparison shows that the proposed idea significantly reduces the test time without increasing the defect level obviously. In addition, the selection method can be computed and updated rapidly according to the statistical data. However, when it comes to actual application, the implementation of machine learning algorithms requires extra time and expertise. The costs and risk associated with the method cannot be ignored. In the future, we will try to apply the method to actual industrial tests to conduct further research. At the same time, we will address the problem of making use of online data, which can update the model in real time to improve the accuracy of prediction. Existing methods in this regard are too computationally expensive to be applied, and this is a problem that we will try to resolve in the future.

Author Contributions: Conceptualization, methodology, J.W. and T.S.; software, W.H.; validation, formal analysis, investigation, resources, writing—original draft preparation, J.W.; writing—review and editing, project administration, funding, H.L. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the Major Scientific Instrument and Equipment Development project under grant no. 62027815, the Foundation of China under the grant no. 61834006.and 2018 Henan Provincial Colleges and Universities Young Key Teacher Training Plan no 2018GGJS219.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Katragadda, V.; Muthee, M.; Gasasira, A.; Seelmann, F.; Liao, J.-H. Algorithm based adaptive parametric testing for outlier detection and test time reduction. In Proceedings of the 2018 IEEE International Conference on Microelectronic Test Structures (ICMTS), Austin, TX, USA, 19–22 March 2018; pp. 142–146.
2. Semiconductor Industry Association. *International Technology Roadmap for Semiconductors*; Semiconductor Industry Association: Washington, DC, USA, 2015.
3. Song, T.; Liang, H.; Sun, Y.; Huang, Z.; Yi, M.; Fang, X.; Yan, A. Novel application of deep learning for adaptive testing based on long short-term memory. In Proceedings of the 2019 IEEE 37th VLSI Test Symposium (VTS), Monterey, CA, USA, 23–25 April 2019.
4. Liu, M.; Pan, R.; Ye, F.; Li, X.; Chakrabarty, K.; Gu, X. Fine-grained adaptive testing based on quality prediction. In Proceedings of the 2018 IEEE International Test Conference (ITC), Phoenix, AZ, USA, 29 October–1 November 2018.
5. Lin, G.-Y.; Tsai, K.-H.; Huang, J.-L.; Cheng, W.-T. A test-application-count based learning technique for test time reduction. In Proceedings of the VLSI Design Automation and Test 2015 International Symposium, Hsinchu, Taiwan, 27–29 April 2015.
6. Song, T.; Liang, H.; Ni, T.; Huang, Z.; Lu, Y.; Wan, J.; Yan, A. Pattern reorder for test cost reduction through improved SVMRANK algorithm. *IEEE Access* **2020**, *8*, 147965–147972. [CrossRef]
7. Li, Y.; Yilmaz, E.; Sarson, P.; Ozev, S.; Yılmaz, E. Online information utility assessment for per-device adaptive test flow. In Proceedings of the 2018 IEEE 36th VLSI Test Symposium (VTS), San Francisco, CA, USA, 22–25 April 2018.
8. Zhan, W.; Shao, Z. Test patterns reordering method based on Gamma distribution. *Integration* **2020**, *72*, 66–71. [CrossRef]
9. Ahmadi, A.; Nahar, A.; Orr, B.; Past, M.; Makris, Y. Wafer-level process variation-driven probe-test flow selection for test cost reduction in analog/RF ICs. In Proceedings of the 2016 IEEE 34th VLSI Test Symposium (VTS), Las Vegas, NV, USA, 25–27 April 2016.
10. Arslan, B.; Orailoglu, A. Aggressive test cost reductions through continuous test effectiveness assessment. *IEEE Trans. Comput. Des. Integr. Circuits Syst.* **2016**, *35*, 2093–2103. [CrossRef]
11. Xue, C.; Shawn, R.D. A one-pass test-selection method for maximizing test coverage. In Proceedings of the 2015 33rd IEEE International Conference on Computer Design (ICCD), New York, NY, USA, 18–21 October 2015; pp. 621–628.
12. Grady, M.; Pepper, B.; Patch, J.; Degregorio, M.; Nigh, P. Adaptive testing-cost reduction through test pattern sampling. In Proceedings of the 2013 IEEE International Test Conference (ITC), Anaheim, CA, USA, 6–13 September 2013.
13. Wang, K.-L.; Lin, B.-Y.; Wu, C.-W.; Lee, M.; Chen, H.; Lin, H.-C.; Peng, C.-N.; Wang, M.-J. Test cost reduction methodology for InFO wafer-level chip-scale package. *IEEE Des. Test* **2017**, *34*, 55–58.
14. Agrawal, M.; Chakrabarty, K. Test-cost modeling and optimal test-flow selection of 3-D-stacked ICs. *IEEE Trans. Comput. Des. Integr. Circuits Syst.* **2015**, *34*, 1523–1536. [CrossRef]
15. Pan, R.; Zhang, Z.; Li, X.; Chakrabarty, K.; Gu, X. Black-box test-coverage analysis and test-cost reduction based on a Bayesian network model. In Proceedings of the 2019 IEEE 37th VLSI Test Symposium (VTS), Monterey, CA, USA, 23–25 April 2019.
16. Zhao, D.; Upadhyaya, S. Dynamically partitioned test scheduling with adaptive TAM configuration for power-constrained SoC testing. *IEEE Trans. Comput. Des. Integr. Circuits Syst.* **2005**, *24*, 956–965. [CrossRef]
17. Semiconductor Industry Association. *International Technology Roadmap for Semiconductors (ITRS)*; Semiconductor Industry Association: Washington, DC, USA, 2012.
18. Chen, M.; Orailoglu, A. Test cost minimization through adaptive test development. In Proceedings of the 2008 IEEE International Conference on Computer Design, Lake Tahoe, CA, USA, 12–15 October 2008; pp. 234–239.
19. Stratigopoulos, H.-G.; Streitwieser, C. Adaptive test with test escape estimation for mixed-signal ICs. *IEEE Trans. Comput. Des. Integr. Circuits Syst.* **2018**, *37*, 2125–2138. [CrossRef]
20. Marinissen, E.J.; Singh, A.; Glotter, D.; Esposito, M.; Carulli, J.M.; Nahar, A.; Butler, K.M.; Appello, D.; Portelli, C. Adapting to adaptive testing. In Proceedings of the 2010 Design, Automation & Test in Europe Conference & Exhibition (DATE 2010), Dresden, Germany, 8–12 March 2010; pp. 556–561.
21. Ng, A.Y.; Jordan, M.I. On discriminative vs. generative classifiers: A comparison of logistic regression and naive bayes. *Adv. Neural Inf. Process. Syst.* **2002**, *2*, 841–848.
22. Wang, L.; Luo, M. Machine learning applications and opportunities in IC design flow. In Proceedings of the 2019 International Symposium on VLSI Design, Automation and Test (VLSI-DAT), Hsinchu, Taiwan, 22–25 April 2019.
23. Yu, L.; Liu, H. Feature selection for high-dimensional data: A fast correlation-based filter solution. In Proceedings of the Twentieth International Conference on Machine Leaning, Washington, DC, USA, 21–24 August 2003; pp. 856–863.
24. Chawla, N.V.; Bowyer, K.W.; Hall, L.O.; Kegelmeyer, W.P. SMOTE: Synthetic minority over-sampling technique. *J. Artif. Intell. Res.* **2002**, *16*, 321–357. [CrossRef]
25. Wang, H. A method of feature selection for continuous features base on similarity degrees of interval numbers. In Proceedings of the 2013 International Conference on Information and Network Security (ICINS 2013), Beijing, China, 22–24 November 2013.
26. Yilmaz, E.; Ozev, S.; Butler, K.M. Per-device adaptive test for analog/RF circuits using entropy-based process monitoring. *IEEE Trans. Very Large Scale Integr. VLSI Syst.* **2013**, *21*, 1116–1128. [CrossRef]
27. Pan, R.; Zhang, Z.; Li, X.; Chakrabarty, K.; Gu, X. Black-box test-cost reduction based on Bayesian network models. *IEEE Trans. Comput. Des. Integr. Circuits Syst.* **2021**, *40*, 386–399. [CrossRef]
28. Liu, M.; Ye, F.; Li, X.; Chakrabarty, K.; Gu, X. Board-level functional fault identification using streaming data. *IEEE Trans. Comput.Aided Des. Integr. Circuits Syst.* **2020**. [CrossRef]