Journal of Electronic Testing (2022) 38:279–288
https://doi.org/10.1007/s10836-022-06009-8

Research on Analog Integrated Circuit Test Parameter Set Reduction Based on XGBoost

Yindong Xiao[1] · Yutong Zeng[1] · Qiong Wu[1] · Ke Liu[1] · Yanjun Li[1] · Chong Hu[2]

Received: 26 December 2021 / Accepted: 6 June 2022 / Published online: 24 June 2022

Abstract
As the scale of integrated circuits continues to increase and their test cost increases with test time, how to optimize the test parameters is an important topic. In analog integrated circuits, the implicit dependency among test parameters makes it possible to apply the XGBoost technique based on decision trees in machine learning to optimize the test parameters. In this paper, an optimization algorithm is proposed based on the XGBoost decision tree model. By modeling the representational relationships of each test parameter in the historical test data set, the list of those to be optimized is obtained according to the descending order of the escape rate in the prediction results. According to this list, the test parameters to be deleted are selected in turn, the prediction results of the remaining test parameters on those test parameters are obtained, and the escape rate after screening out the target parameters is evaluated, and the test parameters are optimized based on this list to reduce the test time and test cost.

Keywords Machine learning · Decision tree · Integrated circuit testing

1 Introduction

With the level of integration and complexity of chips having increased significantly, their cost is getting much more attention. Due to the limitation of chip integration, Automatic Test Equipment(ATE) is used to automate the testing process. However, as the level of integration improves, required testing time grows with it, and the use of ATE is billed by usage time. As for that, the reduction of testing time means the reduction of the overall cost of the test. It has been proved that optimizing the test parameters or adopting a new test mode can effectively reduce the test time [10, 28] Brockman and Director [7] proposed a method to reduce the number of test parameters needed to verify all performances by mining the correlations between test parameters in 1989. Milor and Sangiovanni-Vincentelli [18] reduce number of tests by select test set according to fault model. A binary decision tree model approach is implemented in [5], which also compressed the test set effectively [1, 12, 19]. Adaptive test optimization methods based on a statistical analysis of data correlation have also gained widespread attention in reducing test time. Benner and Boroffice [2] proposed an adaptive test method based on a statistical method to determine whether the test parameters should stay, in [27] the test set is adjusted separately according to the characteristics of each device, [26] schedules tests based on offline statistics obtained from the circuit and instant statistics from each chip under test, reduces test time with effect while keeping the escape rate virtually constant.

Not only the number of test parameters has great importance in test time, but also the sorting of them. Detecting as

Responsible Editor: M. Sachdev

✉ Yanjun Li
yjli@uestc.edu.cn

Yindong Xiao
xydarcher@uestc.edu.cn

Yutong Zeng
525068970@qq.com

Qiong Wu
sherry31597@icloud.com

Ke Liu
liuke@uestc.edu.cn

Chong Hu
hucong@guet.edu.cn

[1] School of Automation Engineering, University of Electronic Science and Technology of China, Chengdu, China

[2] Guangxi Key Laboratory of Automatic Testing Technology and Instruments, Guilin University of Electronic Technology, Guilin, China

many faults as possible at an early stage can greatly optimize test time. A two-phased adaptive test method based on the correlation between parameters is proposed in [8] for parameter test, where the test items successfully reduced in the first stage are ranked in the second stage to maximize the overall probability of early fault detection. In [14] they don't reduce the number of test parameters, but only use the fault coverage information generated by the samples to perform a heuristic ranking of the test parameters, which also significantly reduces the test time. Li et al. [15] proposed a way to reduce test time by using mutual information to evaluate the utility of the next test in the queue of the fault detection circuit, based on the interrelationship between the next and current performed test. They also proposed a new sorting algorithm for the test list in [16], which allows a larger amount of information to be obtained in the early stages of the test, thereby further reducing the test time and error rate. However, the adaptive test optimization method is mainly based on Bayesian theory to optimize the values of different test parameters of a single sample. The optimized test time is uncertain, which will cause difficulty in scheduling in practical applications.

The analog integrated circuit has great nonlinearity in theory. In the manufacturing and testing process, the accurate prediction of this type of nonlinear interval can reduce the number of test parameters required for the test, thereby reducing the test cost. Machine learning technology has unique advantages in nonlinear data fitting. In recent years, machine learning has been widely used in test compression, fault diagnosis, outlier detection, etc. [20] and has achieved great results. As mentioned in [22, 24, 25], the use of machine learning methods to assist chip design, test set compression, and test plan design, already has relatively mature applications. The experimental comparison based on the production test data of a single radio frequency device in [21] also shows that the more complex and advanced machine learning algorithm ontology neural network compared with the traditional direct maximum coverage algorithm or more simple nearest neighbor algorithm and linear discriminant analysis has a greater advantage in reducing the test time of radio frequency integrated circuits. In [13], the minimum redundancy and maximum correlation algorithm are used to select important test parameters and put the data into the BP feedback neural network to use some test parameters to predict the faulty chips, which effectively reduces the test cost and escape rate.

In this paper, a machine learning method based on XGBoost [9] is proposed. An independent model is established for each test parameter, and the ability of the test parameter to be jointly predicted by other test parameters is evaluated, and according to the predictive ability, the priority of optimizing test parameters is determined. We use the key parameter in the analog IC to test the escape rate, namely Defective Parts Per Million(DPPM), as the main criterion of the model. And the corresponding loss rate is defined as the number of lost products per million (Loss Parts Per Million, LPPM) to balance the aggressive strategy in the training process and reduce the probability of overfitting. This method is optimized according to the test parameters, and the result has the characteristic of a fixed test time, which is suitable to apply in the actual analog IC production process.

This paper is structured as follows: Sect. 2 will give a general and concise description of XGBoost and the hyperparameter optimization principle. Section 3 will detail the data preprocessing methods and training process. Section 4 will show the experiment process, results, and analysis for it, followed by the conclusion and prospect in Sect. 5

2 Basic Theory

The purpose of analog integrated circuit testing is to determine whether the test parameters of the tested chip meet the product specifications and whether it is a good or defective product. If the test value or test conclusion of a test parameter in all tested chips can be accurately predicted, then this test parameter can be deleted from the test process, thereby deleting test parameters can reduce the test time.

Machine learning uses historical data to make predictions about samples in different situations, and its models are mainly classified into two types: regression and classification. The regression model mainly provides the numerical fitting ability of the sample, and the classification model mainly completes the label prediction of the sample. The purpose of integrated circuit testing is to determine whether the chip under test meets the product specification requirements and should be classified as a classification problem. Since the method to be implemented involves numerous parameter features, long training time and high accuracy requirements, XGBoost decision tree is chosen as the base model.

The above two models can all be used in the optimization of IC test parameters. Assuming that a certain test parameter is expected to be optimized, a regression model can be trained to learn the relationship between the remaining test parameters and the target parameter, and predict the value of this parameter for all chips, check whether the predicted value meets the product specifications, finally verify its influence on the test conclusion. If a classification model is used, the remaining test parameters need to be used to directly predict the test conclusion, and the algorithm performance is judged based on the prediction accuracy of the test conclusion. As the method of using the regression model to predict the test parameters of analog ICs has better interpretability and can further reveal the relationship between test

parameters, this paper is mainly based on this type of model for in-depth research.

On the other hand, XGBoost algorithm is an excellent general learning algorithm. This paper uses this algorithm as a basis to expand its application in the analysis of analog IC test parameters sets. It should be pointed out that in the machine learning algorithm, the tested chip is called a sample, its test parameter is referred to like the feature, and the test conclusion is called the label.

2.1 XGBoost

XGBoost is a decision tree based model.A common structure of a decision tree model is shown in Fig. 1, where each non-leaf node represents a feature and a leaf node represents a label or decision outcome.a branch represents a judgment condition. When applying this decision tree, the sample to be predicted is checked for a decision condition by the feature value corresponding to the node, which determines the next node position and decision condition, and so on and so forth until a definite decision result is obtained.

which selects features that are significant for the task based on the current learning objective, i.e., for distinguishing between different item properties, as the root node, searches for the feature with the maximum information entropy among the remaining features as the next child node, and repeats in this order until a structure of attributes satisfying the initial classification requirements is constructed.

Information entropy is a common metric used to measure the purity of a sample set and is defined as:

$$Ent(D) = \sum_{k=1}^{|y|} p_k \log_2 p_k$$

where $p_k(k = 1, 2, \ldots, |y|)$ is the proportion of samples of class k in the current sample set D. The smaller the value of information entropy, the higher the "purity" of the sample set, that is, the samples in the sample set of the node belong to the same class as much as possible. Information gain is defined as:

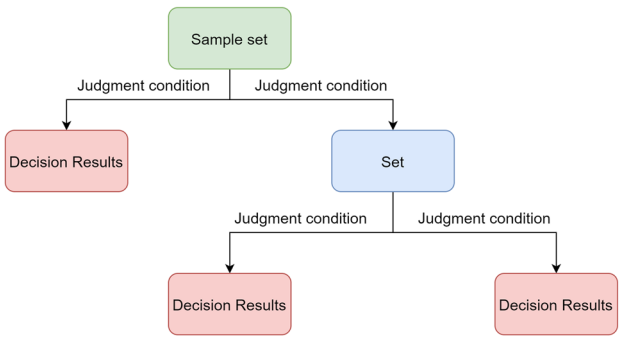

Fig. 1 Decision Tree Model

$$Gain(D, a) = Ent(D) - \sum_{v=1}^{V} \frac{|D^v|}{|D|} Ent(D^v)$$

where D^v is the sample where the vth branch node contains all attributes in D that take the value a^v on a, $\frac{D^v|}{|D|}$ is the weight of the branch node, which means that the higher the number of samples the higher the impact caused by the branch node. The higher information gain means the higher the purity obtained by dividing with attribute a. The information gain is calculated at each branch node to select the next branch and finally a complete decision tree is obtained.

Although the simple decision tree structure is simple and logical, it is prone to overfitting, i.e., the trained model performs very well for the training set but performs poorly for the prediction of unknown samples. Therefore, pruning and random forest-based methods have been derived to mitigate overfitting. Random forests use bootstrap sampling to construct multiple random training sets and respectively train a weak learner decision tree based on them, and then aggregates these decision trees into a strong learner decision tree with significantly superior performance to avoid over-fitting, but the integrated decision trees are independent of each other and have no feedback effect with each other, which derives the gradient descent decision tree method (Gradient Boosting Decision Tree, GBDT).

Each tree in GBDT is fitted using the residuals (i.e., observations of the error) from the previous tree, and the final result is determined by the sum of the results of all trees, but this feature also makes it impossible to perform parallel operations. The main improvement of XGBoost over GBDT is the addition of a regularization term in the cost function to reduce overfitting, and the use of column subsampling in random forest [6], which not only further reduces the risk of overfitting, but also accelerates the parallel computation.

2.2 Hyperparameter Tuning

The parameters obtained in the process of model training are called model parameters, and those whose ranges or values are specified before the model is established are called hyperparameters, which are used as guidance for model training. For example, the learning rate and depth of the tree in the decision tree model training are all hyperparameters. The setting of hyperparameters will affect the performance of the model, and more accurate prediction results can be obtained by adjusting them. The current mainstream auto-tuning algorithms include random search and Bayesian optimization.

HyperOptSeach is a search algorithm supported by the HyperOpt library [4]. The tree-structured Parzen Estimators (TPE) algorithm [3] implemented in it has the characteristic of high accuracy of the output model. It's an algorithm based

on Sequential Model-based Global Optimization(SMBO), which models the conditional probability model $p(x, y)$ of loss function y and hyperparameter x based on the Gaussian process. We use two density functions to define:

$$p(x,y) = \begin{cases} l(x) & \text{if } y < y^*, \\ g(x) & \text{if } y > y^*. \end{cases}$$

these two functions use observed samples and remaining samples to construct probability density functions separately, and y^* is the threshold of loss. $\frac{g(x)}{l(x)}$ can be obtained after fitting $g(x)$ and $l(x)$, and taking $\frac{g(x)}{l(x)}$ as the optimization target to minimize it can we get the expected maximum x^*.

In the model training process, all hyperparameters are tuned by the automatic tuning algorithm, and the search basis is determined by the loss function.

3 Methodology

The main means to optimize the test process of analog ICs in this paper is to reduce the number of parameters that need to be tested as much as possible while ensuring the test conclusion constant. This goal is equivalent to searching for the smallest feature set that can correctly predict the test conclusion in a machine learning problem. The search process is a problem with the complexity of A. To speed up the problem-solving process, the method we proposed is carried out according to the steps shown in Fig. 2

As is shown in Fig. 2, the data pre-processing procedure is to normalize test parameters according to product specifications. A single parameter scan will establish a learning model for each test parameter, use the remaining test parameters to predict the corresponding target parameter, and obtain its loss function value. Sort all test parameters according to the loss function value, we can get the list of test parameters to be deleted. The test parameters are sequentially deleted in the order of the list, and the deleted parameters will be modeled and predicted using the remaining test parameters after each deletion, and a new model set can be obtained. Lastly, evaluate the impact of the models in the set on the test conclusions.

Therefore, there are two aspects needed to be focused on: 1. Define the model loss function unique to the simulated IC test dataset as the optimization target; 2. Select a parameter that can reflect the characterization ability of the test parameter to be deleted as the sorting basis.

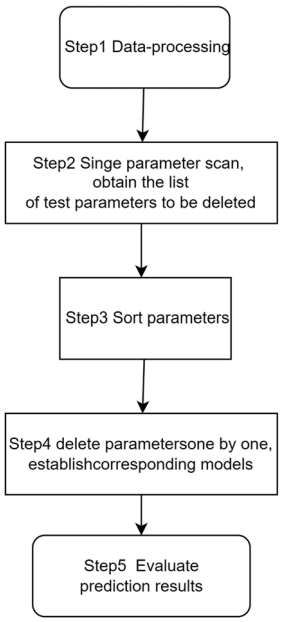

Fig. 2 Flow Chart of the Optimization Algorithm

3.1 Loss Function

The loss function is used to define the degree of inconsistency between the predicted value and the true value of the model, it determines the effect of the training model. Choose the composition of the loss function according to the application characteristics of the analog IC:

$$loss = DPPM + \alpha LPPM$$

where DPPM is the number of defective units per million units, which reflects the escape rate of the test, which is calculated as:

$$\frac{n^{'}}{n} * 1000000$$

where $n^{'}$ represents the number of defective products, and n represents the total number of products with a good test result.

LPPM, Loss Parts Per Million, is the attrition rate, we define it as the number of good products among the products whose test conclusion is defective, it reflects the model's misjudgment rate of good products. This is the loss caused by the application of the optimization algorithm. Adding this part of the parameter to the loss function can prevent the learning algorithm from predicting a large number of good products as defective products in order to reduce the test escape rate.

α is a cost ratio that responds to the acceptable level of attrition, i.e., the level of acceptability of discarding good products. When $\alpha < 1$ indicates that the model places more emphasis on reducing the escape rate, therefore, in this paper, the empirical value $\alpha = 0.5$ emphasizes the importance that the model gives to the escape rate.

It is worth noting that the XGBoost algorithm has a series of built-in objective functions, which are the sum of the loss function and the regularization term, and do not support

custom objective functions. As we point out before, the purpose of model training is to obtain a fit to those target test parameters. Therefore, when training a single model, an objective function based on Mean Squared Error (MSE) is selected. The loss function defined in this section is mainly used for model evaluation during hyperparameter tuning.

3.2 Model Training Process

The most important objective of this overall model training is to obtain the best ranking of the features for subsequent redundant term removal.

As shown in Fig. 3, the overall training process of the model is divided into four main parts.

① : Set all features as an overall feature set $A = \{a_i\}, i = 1, 2, \ldots, n$, n is the total number of test parameters, a_i is the measurement value set of the i-th test parameter of all tested chips, which represents a single test parameter.The test parameter a_i is separated from A,and the remaining parameter sets form a new feature set $A^{'} = \{a\_j\}, j = 1, 2, \ldots, n, j \neq i$.

② : Perform regional model training on all $i = 1, 2, \ldots, n$, obtain the prediction models of the remaining test parameter set $A^{'}$ on the target parameter a_i, and the related loss function value $loss_i$.

③ : Sort the test parameters according to $loss_i$,.and obtain $S = \{s_i\}, i = 1, 2, \ldots, n$,which is the list of test parameters to be deleted, where s_i is the sorted test parameter.

④ : Divide S into parameter set $T_k = \{s_i\}, i = 1, 2, \ldots, k$, and remaining parameter set $R_k = \{s_j\}, j = k + 1, \ldots\ n$, take T_k as the target, and then train the model whose input is R_k. Then models can be obtained for $k = 1, 2, \ldots, n$. Their corresponding DPPM and LPPM can reflect the influence of the optimized model on the test conclusion.

Based on the predicted results, the analogue IC test engineer can select the number of test parameters to be optimised according to the test escape rate and loss rate.

3.3 Hyperparameters

The hyperparameters that need to be set before training are divided into General Parameters, Booster Parameters, and Learning Task Parameters. In this paper, according to the type, the parameters that need to be set are General Parameters: 1. Booster, which indicates the type of weak learner applied, can choose three models: linear model gblinear, tree model dart [23], gbtree; 2. num_threads, which indicates the number of parallel threads in the training process.

Booster parameters: 1. learning_rate, also available as eta, represents the learning rate; 2. min_split_loss, also available

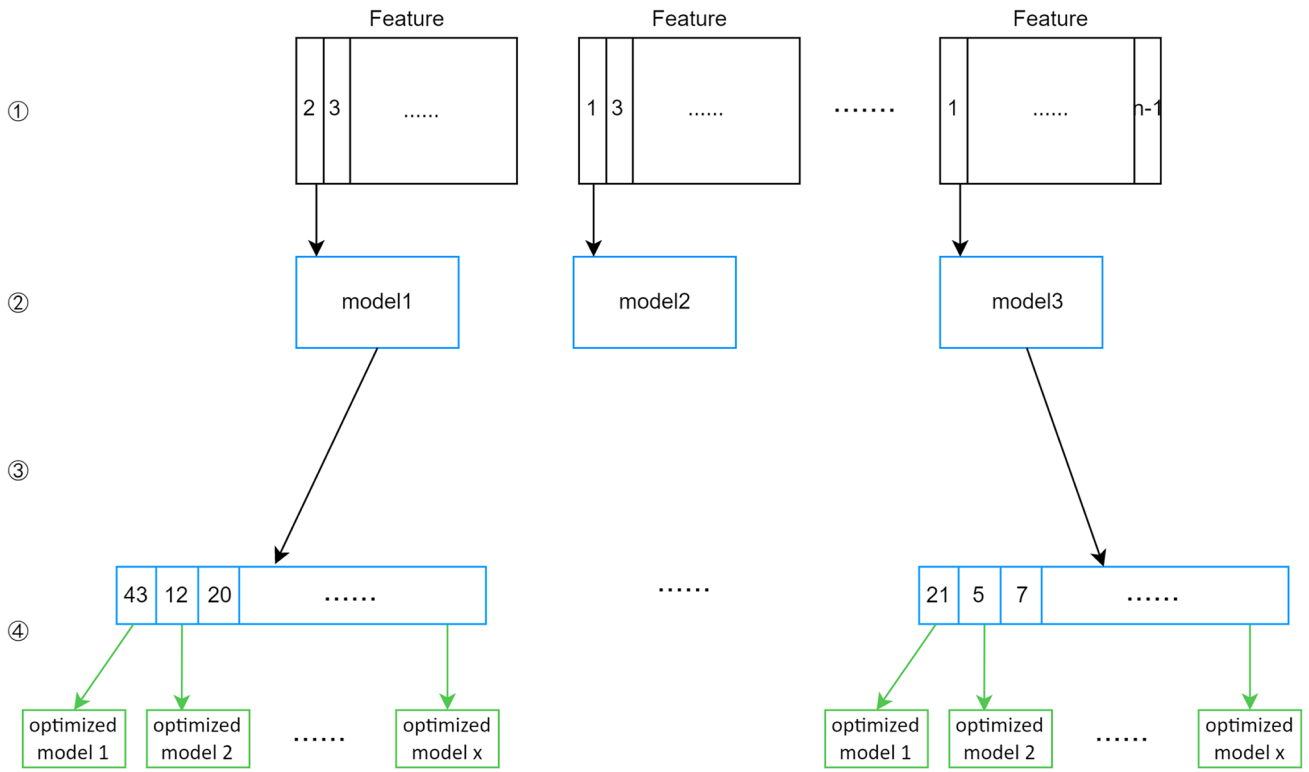

Fig. 3 Schematic Diagram of Model Establishment

as gamma, represents the minimum loss value required for the tree to perform the next branching operation; 3. max_ depth, represents the maximum depth of the tree model; 4. min_child_ weight, denotes the sum of the instance weights required for the smallest leaf node, if the sum of the instance weights of the leaf nodes generated during further division of the tree is less than min_child_weight then the tree does not generate new branches; 5, subsample, denotes the proportion of subsamples of training instances; 6, colsample_ bytree, with colsample_bylevel and colsample_bynode, is a set of parameters that control the secondary sampling of columns, which controls the proportion of subsamples of the columns when each tree is constructed; 7. grow_policy, which controls the method of adding new nodes to the tree, with the option of branching from the node closest to the root node (depthwise) and branching at the node with the largest change in loss (lossguide).

Learning Task Parameter: objective: defines the objective function to be minimized, we can choose from reg:squared error, reg: logistic, etc.

The configuration range of hyperparameters is shown below Table 1:

4 Experiment, Results and Analysis

In this section we explicit the experimental environment, preparation for the data set, the model training results, and the analysis of the experimental results.

4.1 Dataset

The test data of integrated circuits usually involve high-level commercial secrets, and there is currently no public dataset available for comparison. The data used in the experiment in this article comes from an analog integrated circuit manufacturer, containing two types of chips, one is an analog circuit with wireless function; the other is an analog circuit with mixed-signal function. The test datasets of the two circuits are referred to as dataset A and dataset B in the subsequent articles.

Table 1 Parameter Setting

Hyperparameter name	Configuration range	type
booster	dart/gbtree	General Parameter
num_thread	Maximum number of threads available	
learning_rate	$[10^{-4}, 1]$	Booster Parameter
min_split_loss	$[10^{-8}, 10]$	
max_depth	[1,11], take integer	
min_child_weight	[1, 8], take integer	
subsample	[0.5, 1.0]	
colsample_bytree	[0.3, 0.7]	
grow_policy	depthwise/lossguide	
objective	reg: squarederror	Learning Task Parameter

Among all the test parameters of a single tested chip in the dataset, if there is a test parameter whose value exceeds the range of product specifications, the chip is classified as a defective product. The types of test parameters include power supply voltage, current, output signal voltage amplitude, offset, etc.

Dataset A contains 80000 chips, and each chip has 70 test parameters. Among them, good products (negative samples), that is chips that passed the test, have a total of 70000; the rest are defective products that failed the test (positive samples), and the yield is about 87.5%. Dataset B contains 118000 chips, and each chip has 42 test parameters. Among them, there are 113000 good products, and the yield is about 95.8%.

4.2 Experimental Environment and Process

Due to the large amount of data, the calculations required for the training process increases, and the requirements for the hardware environment are also higher. Therefore, this experimental environment uses both GPU and CPU. We use 2 Intel Xeon E5-2673 v3 CPUs with a main frequency of 2.40GHz, 4 GTX 1080 Ti graphics cards from NVIDIA, and 128GB DDR4 SDRAM high-bandwidth memory to set up the experimental platform.

The pre-processing of data is crucial before the training of the learning machine. The wafer's test data is stored in matrix form, each row represents a chip, and each column represents a test parameter. Since these data involve commercial secrets, they need to be normalized first and then they can be used for model training and learning progress. The data within the range of the product specification is normalized to the closed interval of [0,1] through linear transformation, and the data beyond the range will fall outside the interval, so the chip can be judged as a defective product. Because part of the test parameters of the tested chip already meets the product specification, therefore they only contain little information and will not cause the tested chip to be judged as a defective product, there is no need for further analysis and optimization of such test parameters, so they can be deleted in advance. After excluding such test parameters, there are only 29 candidate test parameters to be analyzed in dataset A and 38 candidate test parameters in dataset B.

To prevent over-fitting, it is impossible to directly use all data to train the model, so the concept of cross-validation is adopted to cross-validate the dataset. This concept was first proposed by Seymour Geisser. The basic idea is to divide

the dataset into a training set and a validation set according to certain rules. The training set is used to train the model, and the validation set is used to test various indicators of the model. In this experiment, to ensure a balanced distribution of positive and negative samples, during the segmentation process of the training set, validation set, and test set, the proportions of good and defective products are the same as the overall ratio.

In our model training, all test chips are divided into training set and validation set according to a 4:1 ratio by Stratified cross-validation. Stratified cross-validation can ensure the balanced distribution of positive and negative samples between the training set and the validation set by stratified sampling. The training set is used for model training, and the validation set is used to verify the generalization and adaptability of the model. At the same time, during the training process, it is necessary to obtain the effect of tuning the hyperparameters on the model. As for that, the algorithm will randomly split part of the test chip data from the training set as the test set, which is specifically used to verify the effect of hyperparameter changes on the objective function. This segmentation and training will be repeated multiple times to ensure the stability of the algorithm, the final objective function will be determined by the model with the largest loss.

The hyper-parameter optimisation of the model is carried out using the pre-processed dataset. As mentioned above, in this experiment, the HyperOptSeach in the distributed reinforcement learning module Ray. tune [17] is used for hyperparameter's autonomous tuning. This module can run multiple sets of hyperparameter configurations in parallel. Since the range of hyperparameter configurations may be large and needs to be tested frequently during the training process, this method will undoubtedly save a lot of time.

After the hyperparameter training is configured, the segmented training set is used for model training. After the overall training of the model is completed, the test set data is put into the model to verify the testing effect after the test parameters are deleted as planed. The reduction of test parameters is the process of eliminating redundant test parameters and also the process of test parameters selection. The information contained in redundant test parameters can be fitted by other test parameters, so they are irrelevant test parameters in the learning task. Feature selection can alleviate the curse of dimensionality [11] to a certain extent while removing a few features usually reduces the difficulty of machine learning problems and improves efficiency.

4.3 Experimental Result

This paper is based on two sets of simulated IC test data for test verification.

Table 2 Single Test parameter Fitting Results of Dataset A

Parameter number	DPPM	LPPM	PPM	MSE
26	0.0	0.0	0.0	1.07×10^{3}
25	0.0	0.0	0.0	1.32×10^{6}
27	0.0	0.0	0.0	8.12×10^{7}
33	0.0	116.4	58.2	2.42×10^{-3}
28	58.2	0.0	58.2	2.36×10^{-2}
32	116.4	0.0	116.4	6.51×10^{-3}
24	116.4	0.0	116.4	7.77×10^{1}

The first step of model training is to use all the remaining test parameters in the training set and fit each test parameter in turn to form a local training model. The test parameters are numbered according to the final censoring order and labeled as parameter numbers, and the DPPM, LPPM, and least squared error (MSE) values of the test parameters in the training set are obtained and arranged in ascending order of loss as defined above. DPPM and LPPM take one valid number after the decimal point, while MSE takes two valid numbers after the decimal point. The results are shown in the table below. Only the first 7 items are shown Tables 2 and 3.

From the results of single test parameter scanning, it can be seen that the MSE of the test parameters has no decisive effect on DPPM and is generally irrelevant. This proves the necessity of choosing different loss functions for hyperparameter tuning in this method.

After performing model training on the two datasets, the variation trends of DPPM and LPPM after subtracting different numbers of test parameters are shown in Figs. 4 and 5 respectively.

4.4 Analysis of Results

We can see that the DPPM had been kept at zero when the number of the deleted parameter was lower than 5 from the experimental results of dataset A. With the number of deleted test parameters increased, the DPPM generally

Table 3 Single Test parameter Fitting Results of Dataset B

Parameter number	DPPM	LPPM	PPM	MSE
58	0.0	0.0	0.0	1.45
57	0.0	42.5	21.25	1.60
62	42.5	0.0	42.5	1.35
64	42.5	0.0	42.5	1.37
60	0.0	127.4	63.7	1.59
46	84.8	0.0	84.8	0.08
37	84.9	42.5	106.15	0.15

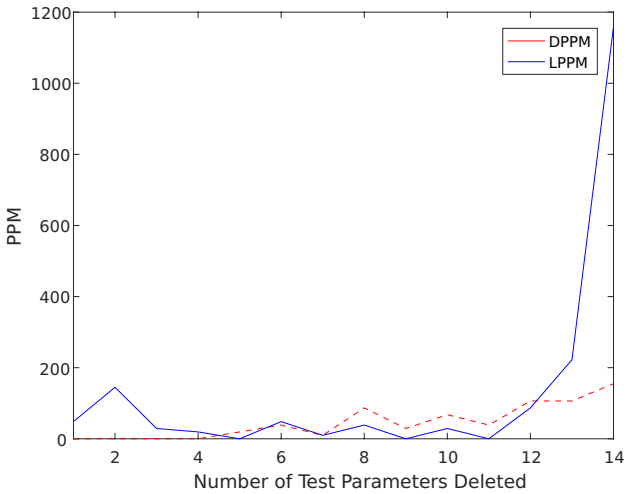

Fig. 4 Schematic Diagram of Model Establishment

showed a fluctuating upward trend. As for LPPM, when deleting the first 2 parameters there was a big fluctuation on its curve. But until the number of deleted parameters reached 11, it maintained a slightly fluctuating low value, after that the LPPM appeared an almost straight upward trend.

Concerning the results of dataset B, the LPPM maintained a rather stable low value when the number of deleted test parameters was less than 11, when the number of deleted parameters was greater than 11, large irregular fluctuations occurred. DPPM also maintained a low value when the number of deleted test parameters was less than 9, but then it increased exponentially.

Reducing the test parameter set can undoubtedly save test time. However, it can be seen from the figures that as the number of deleted test parameters increases, the number

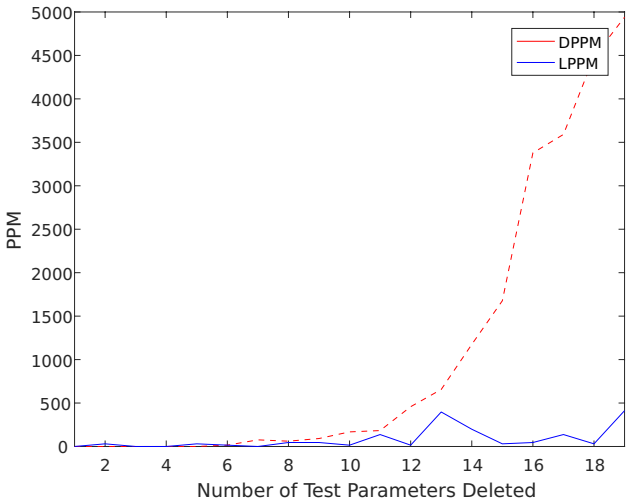

Fig. 5 Schematic Diagram of Model Establishment

of defective chips in the products whose test conclusion is good also increases exponentially, good products misjudged as defective ones are also increasing. In the meantime, we can see from the experimental result of dataset A, when the deleted test parameters are too few, LPPM is relatively large. This happens because we set to emphasize the influence of DPPM on the model. Therefore, when selecting the final deletion test parameters, it is necessary to make a joint decision based on DPPM and LPPM.

If we set the acceptable test escape rate under 20PPM, the test parameters that can be deleted in the two sets of data can reach 5 and 6, accounting for 17.24% and 15.79% of all analyzable parameters respectively.

5 Summary and Outlook

The method we proposed in this paper can provide a basis for deleting the test parameters of analog integrated circuits and can reduce the cost of time and money for analog IC testing. Experiments show that its performance is outstanding and can provide no more than 15.79% test parameter optimization. At the same time, as the method can optimize test parameters statically, it has a better scope of application than adaptive tests.

Meanwhile, the method described in this article needs to be retrained when applied to products of different manufacturers and designs because different processes in the production process have different effects on-chip parameters. This is also the limitation of methods based on the data optimization method.

The core of the method is the test parameter censoring priority ranking, which is one of the research areas in the direction of feature engineering in machine learning. There is still room for exploration in this direction, and it can be further optimized in the future with the help of interpretability measures in machine learning.

Acknowledgments This work was supported by the National Natural Science Foundation of China under Grant 61871089 and Key Laboratory of Automatic Testing Technology and Instruments under Grant YQ201202.

Data Availability The data that support the findings of this study are available from [Xi'an Microelectronic Technology Institute] but restrictions apply to the availability of these data, which were used under license for the current study, and so are not publicly available. Data are however available from the authors upon reasonable request and with permission of [Xi'an Microelectronic Technology Institute].

Declarations

Conflicts of Interests The authors declare there is no conflicts of interest regarding the publication of this paper.

References

1. Amati L, Bolchini C, Salice F (2011) Optimal test set selection for fault diagnosis improvement. In: Proc. IEEE International Symposium on Defect and Fault Tolerance in VLSI and Nanotechnology Systems, pp 93–99
2. Benner S, Boroffice O (2001) Optimal production test times through adaptive test programming. In: Proceedings International Test Conference 2001 (Cat. No. 01CH37260), IEEE, pp 908–915
3. Bergstra J, Bardenet R, Bengio Y, Kégl B (2011) Algorithms for hyper-parameter optimization. Adv Neural Inf Process Syst 24
4. Bergstra J, Yamins D, Cox D (2013) Making a science of model search: Hyperparameter optimization in hundreds of dimensions for vision architectures. In: Proc. International Conference on Machine Learning, PMLR, pp 115–123
5. Biswas S, Blanton RD (2006) Statistical test compaction using binary decision trees. IEEE Des Test Comput 23(6):452–462
6. Breiman L (2001) Random forests. Mach Learn 45(1):5–32
7. Brockman JB, Director SW (1989) Predictive subset testing: Optimizing IC parametric performance testing for quality, cost, and yield. IEEE Trans Semicond Manuf 2(3):104–113
8. Chen M, Orailoglu A (2008) Test cost minimization through adaptive test development. In: Proc. IEEE International Conference on Computer Design, pp 234–239
9. Chen T, Guestrin C (2016) Xgboost: A scalable tree boosting system. In: Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining, pp 785–794
10. Chen W (2014) The optimization study about improving the IC testing yield and reduce the testing time. Master's thesis, University of Electronic Science and Technology of China
11. Hastie T, Tibshirani R, Friedman J, Franklin J (2005) Reviews-the elements of statistical learning: data mining, inference and prediction. Math Intell 27(2):83–84
12. Higami Y, Saluja KK, Takahashi H, Kobayashi Sy, Takamatsu Y (2006) Compaction of pass/fail-based diagnostic test vectors for combinational and sequential circuits. In: Proc. Asia and South Pacific Conference on Design Automation, 2006., IEEE, pp 6–pp
13. Hou W, Liang H, Song T et al (2021) Integrated circuit test method based MRMR algorithm and BP neural network. Microelectronics 51(5):766
14. Jiang W, Vinnakota B (2001) Defect-oriented test scheduling. IEEE Transactions on Very Large Scale Integration (VLSI) Systems 9(3):427–438
15. Li Y, Yilmaz E, Sarson P, Ozev S (2018) Online information utility assessment for per-device adaptive test flow. In: Proc. IEEE 36th VLSI Test Symposium (VTS), pp 1–6
16. Li Y, Yilmaz E, Sarson P, Ozev S (2019) Adaptive test for RF/analog circuit using higher order correlations among measurements. ACM Transactions on Design Automation of Electronic Systems (TODAES) 24(4):1–16
17. Liaw R, Liang E, Nishihara R, Moritz P, Gonzalez JE, Stoica I (2018) Tune: A research platform for distributed model selection and training. arXiv preprint arXiv:180705118
18. Milor L, Sangiovanni-Vincentelli A (1994) Minimizing production test time to detect faults in analog circuits. IEEE Trans Comput Aided Des Integr Circuits Syst 13(6):796–813. https://doi.org/10.1109/43.285252
19. Shukoor MA, Agrawal VD (2009) A two phase approach for minimal diagnostic test set generation. In: Proc. 14th IEEE European Test Symposium, pp 115–120
20. Stratigopoulos HG (2018) Machine learning applications in IC testing. In: Proc. IEEE 23rd European Test Symposium (ETS), pp 1–10
21. Stratigopoulos HG, Drineas P, Slamani M, Makris Y (2007) Non-RF to RF test correlation using learning machines: A case study. In: Proc. 25th IEEE VLSI Test Symposium (VTS'07), pp 9–14
22. Stratigopoulos HG, Drineas P, Slamani M, Makris Y (2009) Rf specification test compaction using learning machines. IEEE Transactions on Very Large Scale Integration (VLSI) Systems 18(6):998–1002
23. Vinayak RK, Gilad-Bachrach R (2015) Dart: Dropouts meet multiple additive regression trees. In: Artificial Intelligence and Statistics, PMLR, pp 489–497
24. Wang L, Luo M (2019) Machine learning applications and opportunities in IC design flow. In: Proc. International Symposium on VLSI Design. Automation and Test (VLSI-DAT), IEEE, pp 1–3
25. Yi M, Song C, Yu J, et al. (2021) An adaptive test method of IC based on random forest. Journal of Zhengzhou University (Engineering Science) 42(4):6
26. Yilmaz E, Ozev S (2008) Dynamic test scheduling for analog circuits for improved test quality. In: Proc. IEEE International Conference on Computer Design, pp 227–233
27. Yilmaz E, Ozev S (2009) Adaptive test elimination for analog/RF circuits. In: Proc. 46th ACM/IEEE Design Automation Conference, pp 720–725
28. Zhang SH (2011) The application of ping-pong mode in IC production test. China Integrated Circuit 20(12):5

Yindong Xiao is an Associate Professor at the University of Electronic Science and Technology, where he received his PhD in 2013. His research interests include arbitrary waveform generation techniques, automatic test system architecture, machine learning based test optimization, and computer network testing techniques.

Yutong Zeng wasborn in Jiangxi province, China, in 1998. He received the bachelor's degree in automation engineering from University of Electronic Science and Technology of China, Zhongshan Institute in 2020. He is currently studying for a master's degree at the School of Automation Engineering, University of Electronic Science and Technology of China.

Qiong Wu was born in Sichuan province, China, in 1997. She received the bachelor's degree in automation engineering from Nanjing University of Aeronautics and Astronautics in 2019. She is currently studying for a master's degree at the School of Automation Engineering, University of Electronic Science and Technology of China.

Ke Liu was born in Sichuan, China, in 1978. He received the MS and PhD degrees from the University of Electronic Science and Technology of China (UESTC), China, in 2003 and 2010, respectively, where he is currently a Professor in the School of Automation Engineering. His research interests include frequency synthesis, measurement and instrument, and signal processing.

Yanjun Li received the PhD degree in Technology and Instrumentation of Test and Measurement from University of Electronic Science and Technology of China (UESTC) in 2009. He is currently an Associate Professor at the College of Automation Engineering, UESTC. He worked in USA as a visiting scholar at Arizona State University from August 2015 to July 2016 and at University of Notre Dame from May 2007 to November 2007. His research interests are in the areas of mixed-signal circuit test, analog circuit fault diagnosis, and well logging technology.

Chong Hu received the BS degree in computer science and technology from Guilin University of Electronic Technology, Guilin, China, and the MS degree in test and measurement technology and instruments from Guilin University of Electronic Technology, Guilin, China, in 2003 and 2006, respectively. He received the PhD degree in test and measurement technology and instruments from Xidian University, Xi'an, China, in 2017. Now, he is a Professor at Guilin University of Electronic Technology. His current research interests include automatic test system, design-for-testability, test optimization techniques for many-core designs, and fault diagnosis.