A Technique to Aggregate Classes of Analog Fault Diagnostic Data Based on Association Rule Mining

Ruslan Dautov
College of Computer Science &
Software Engineering,
Shenzhen University,
Shenzhen, China
Email: dautovri@szu.edu.cn

Sergey Mosin
Institute of Computational Mathematics &
Information Technologies,
Kazan Federal University,
Kazan, Russian Federation
Email: smosin@ieee.org

***Abstract*—Analog circuits are widely used in different fields such as medicine, military, aviation and are critical for the development of reliable electronic systems. Testing and diagnosis are important tasks which detect and localize defects in the circuit under test as well as improve quality of the final product. Output responses of fault-free and faulty behavior of analog circuit can be represented by infinite set of values due to tolerances of internal components. The data mining methods may improve quality of fault diagnosis in the case of big data processing. The technique of aggregation the classes of fault diagnostic responses, based on association rule mining, is proposed. The technique corresponds to the simulation before test concept: a fault dictionary is generated by collecting the coefficients of wavelet transformation for fault-free and faulty conditions as the preprocessing of output signals. Classificator is based on *k*-nearest neighbors method (*k-NN*) and association rule mining algorithm. The fault diagnostic technique was trained and tested using data obtained after simulation of fault-free and faulty behavior of the analog filter. In result the accuracy in classifying faulty conditions and fault coverage have consisted of more than 99,09% and more than 99,08% correspondingly. The proposed technique is completely automated and can be extended.**

I. INTRODUCTION

Quality and reliability is important factors for efficient development in the microelectronics industry. Tests in collaboration with fault diagnostics play a key role in the process of manufacturing defect localization, detecting reasons for its appearances whilst preparing the data for technological process adjustment thus to increase the yield for the final batch.

Nowadays, manufacturing of analog and mixed-signal integrated circuits are developed very actively. Testing and fault diagnostics for such kind of integrated circuits (IC) are essentially more complex in comparison with digital IC due to the following features: 1) Continuous character of analog signals processing; 2) Nonlinearity and complex functional dependence between the input and output signals; 3) Influence of component tolerance on the value of output signals; 4) High sensitivity of output functions to the deviation of internal component parameters and external environmental parameters; 5) The lack of effective models for defects and faults for analog circuits, etc.

Traditionally, diagnostics of analog circuits are implemented using which here on will be referred as Fault Dictionary (FD), each row of which contains the upper and lower boundaries of the range of possible values for controlled parameters in different test nodes for all considered states of the circuit, i.e. fault-free and faulty states containing different kinds of faults. Fault detection occurs during the output response measurement of the circuit-under-test (CUT) and sequential comparison value is obtained within the boundaries in FD rows. The condition of the CUT is diagnosed when the measured value lays in the boundary range of the corresponding row in FD.

The technique to construct the generalization fault dictionary based on artificial neural network taking into account the component tolerances and using the association rule mining as the preprocessing of a big volume of overlapped data which is proposed in the paper. Proposed technique reduces complexity of fault detection due to associative mode of operation as well as decreases the high size of the FD thanks to implementation of the FD as artificial neural network with fixed architecture for different number of considered faults. Algorithms which are used in this technique are parallel and ready to run on the clusters.

This paper is organized as follows. Section II introduces the background of the proposed technique to the construction of the FD. Mathematical methods used for the main steps of DFT-flow are described in Section III. Experimental results and corresponding analysis are presented in Section IV. Finally, summary and conclusions of this paper are presented in Section V.

II. DESIGN-FOR-TESTABILITY AND FAULT SIMULATION

The concept of design-for-testability (DFT) is widely used nowadays for improving the development process of reliable and easy testable CUT as well as reducing the total time on design and test of the developed circuit [1].

The involvement of the DFT-technique at early stages of a device development is extremely important for the implementation of highly reliable analog and mixed-signal integrated circuits with the guaranteed quality. It provides the principal changes to improve circuit in minimum time and cost overheads.

Test generation is one of the main stages realized in *DFT*-technique, which provides a selection of controlled parameters,

test nodes, and test stimuli for a designed circuit, FD construction with efficiency estimation for obtained test patterns.

The fault simulation is an important task for constructing the FD and next fault diagnostics with high quality. The catastrophic and parametric faults of analog circuits are vastly distinguished. The catastrophic fault is the reason of short or open effects in the circuit whilst parametric fault is caused by deviation of component parameter outside the tolerance. However, the set of catastrophic faults in analog circuit is finite while the set of parametric faults is infinite. High computational effort and the lack of realistic fault models are the main problems of the fault simulation for analog circuits. Component tolerances provide the set of possible responses for each fault, which can be partially or completely overlapped with the set of responses for the fault-free case. Therefore fault simulation is a crucial task to estimate the influence of different faults on the behavior of a correct operating IC.

The parallel computing system allows the performing decomposition of the fault simulation task according to parallel paradigm [2]. The main idea here is the use of independent computational resources for simultaneous simulation of several faults. Essential computational complexity deals with the circuit simulation with injected faults using Monte-Carlo method [3], [4]. The number of Monte-Carlo iterations should be at least thousand or ten thousand times in order to adequately estimate the influence of component tolerances on the fault-free and faulty circuit's behavior. The explicit set of output responses for fault free and faulty conditions are generated as result of simulation which represents the dataset for further big data mining.

In general, the sets of controlled output responses or their essential characteristics (S_i) obtained for different circuit's conditions may partially or completely intersect as well as consisting of values, which do not intersect with values from other sets. From two up to N+1 sets may participate in the intersection, where N is the number of considered faulty conditions for the circuit and one corresponds to the fault-free condition. So, the following subsets can be picked out in the result of the intersection: independent subsets (*IS*), ambiguous double subsets (*ADS*), ambiguous triple subsets (*ATS*) and potentially up to (N+1)-fold ambiguous subset. Eq. (1)-(8) describe the rules of generating the each type of subsets represented in Figure 1.

According to Eq. (1)-(8) , the values from independent subsets IS_i provide definite diagnosis of the *i*-th circuit's condition. Meanwhile, the subsets ADS_{ij} detect the faults *i* and *j* accurately within the ambiguous group *AG* = {*i*, *j*} and subset ATS_{ijk} detects the faults *i*, *j* and *k* accurately within the ambiguous group *AG* = {*i*, *j*, *k*}.

The partial intersection of the sets S_i as well as boundary values and values near boundary between intersected sets provide essential indeterminacy at generalization and may cause the inaccuracy of training and consequently may be the reason of low-quality fault diagnostics of the CUT with alpha and beta errors.

The technique is to training the machine learning model using the subsets obtained after intersection of sets S_i (i = 1..N+1) instead of straight S_i, which allows reducing the indeterminacy at the training stage and increasing the accuracy of further fault diagnosis, which is proposed in this paper.

$$IS_1 = S_1 \backslash S_2 \backslash S_3 \backslash S_4 = \{x \mid x \in S_1 \land x \notin S_2 \land x \notin S_3 \land x \notin S_4\}, \quad (1)$$

$$IS_2 = S_2 \backslash S_1 \backslash S_3 \backslash S_4 = \{x \mid x \notin S_1 \land x \in S_2 \land x \notin S_3 \land x \notin S_4\}, \quad (2)$$

$$IS_3 = S_3 \backslash S_1 \backslash S_2 \backslash S_4 = \{x \mid x \notin S_1 \land x \notin S_2 \land x \in S_3 \land x \notin S_4\}, \quad (3)$$

$$IS_4 = S_4 \backslash S_1 \backslash S_2 \backslash S_3 = \{x \mid x \notin S_1 \land x \notin S_2 \land x \notin S_3 \land x \in S_4\}, \quad (4)$$

$$DS_{12} = (S_1 \cap S_2) \backslash S_3 \backslash S_4 = \{x \mid x \in S_1 \land x \in S_2 \land x \notin S_3 \land x \notin S_4\}, \quad (5)$$

$$ADS_{13} = (S_1 \cap S_3) \backslash S_2 \backslash S_4 = \{x \mid x \in S_1 \land x \notin S_2 \land x \in S_3 \land x \notin S_4\}, \quad (6)$$

$$ADS_{23} = (S_2 \cap S_3) \backslash S_1 \backslash S_4 = \{x \mid x \notin S_1 \land x \in S_2 \land x \in S_3 \land x \notin S_4\}, \quad (7)$$

$$ADS_{123} = S_1 \cap S_2 \cap S_3 = \{x \mid x \in S_1 \land x \in S_2 \land x \in S_3\}. \quad (8)$$

III. Proposed technique for fault diagnostic

New methods of testing a complex circuit require large computing power. Not only computing resources but also resources of different memory levels and communication resources are required. Graphics processing and co-processors units can alleviate the processor bottleneck, but memory or disk bottlenecks can only be eliminated by splitting data across multiple nodes. Multi-nodes computing provides scalable power, so it can eliminate bottlenecks in all three traditional computing

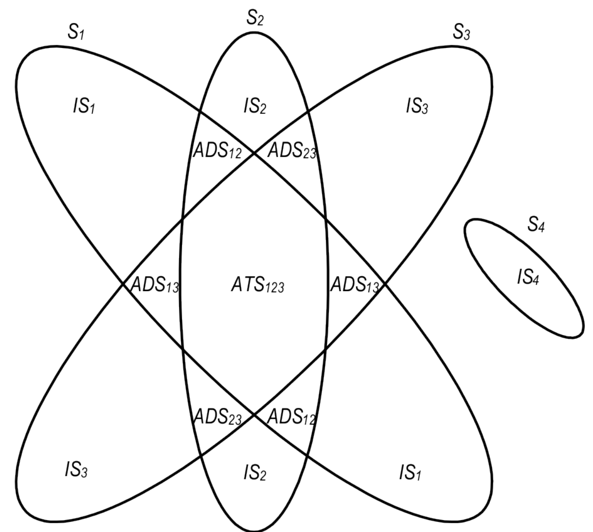

Fig. 1. Subsets generated by the intersection of the fault sets

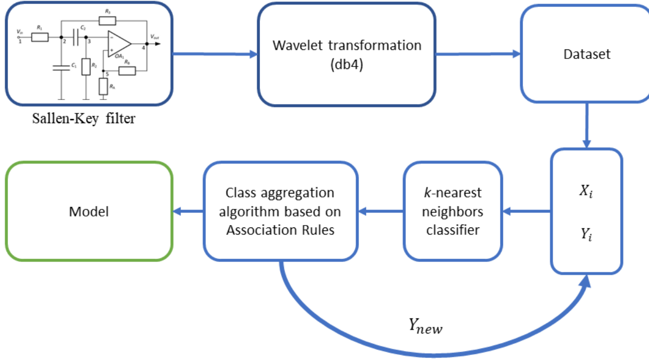

Fig. 2. Main technique steps

resources (computation, memory, communication). However, computationally complex Monte-Carlo simulation can be realized on different nodes with generating an exhaustive large amount of datasets about fault-free and faulty behavior of CUT according to the full concept of big data.

The proposed technique can be described in the following set of main steps:

1) Fault simulation using Monte-Carlo analysis taking into account the component tolerances.
2) Wavelet-decomposition of CUT's output responses.
3) Class Aggregation based on *k*-nearest neighbors algorithm (*k-NN*) and Association Rule algorithms
4) Building Machine Learning Model

A. Monte-Carlo Fault simulation and Wavelet-decomposition

The simulation of analog circuit behavior in the time domain is based on solution the system of differential equations

$$O = F\left(\frac{\partial \mathbf{P}}{\partial t}, \mathbf{P}, t\right), \quad (9)$$

where $\mathbf{O}$ is a vector of output characteristics, $\mathbf{P} = \{p_k\}, k \in N$ is a set of component parameters and t is a time variant.

The random values $\mathbf{P}_i = \{p_k^i\}$ obtained taking into account the component tolerances are used in Eq. 9 instead of the nominal values for each i-th iteration of the Monte-Carlo simulation.

The random value of a parameter p is calculated according to Eq. 10

$$p = p_0(1 + \xi\Delta) \quad (10)$$

where p_0 is a nominal value; ξ is a random centered value on the range [–1, 1] with specified distribution law; Δ is the relative deviation.

The Monte-Carlo simulation is performed for the fault-free circuit as well as for the circuit with injection faults from the fault list.

A sample output signal measured during one period after finishing the transient processes is accumulated as the result of each iteration of the Monte-Carlo simulation. Finally, the set $\mathbf{O}$ of samples for total number s of Monte-Carlo iterations is generated. The use of instant samples for testing and fault diagnosis is not effective due to phase shift, noise and distortion, effect of nonlinearity, etc. Therefore the extraction of essential characteristics for sampled output signals based on transformation from time to frequency domain is proposed.

The discrete wavelet transformation (DWT) is used for extraction the essential characteristics according to Eq. 11

$$W_\Psi(s,\tau) = \int_{-\infty}^{+\infty} O(t)\Psi_{s,\tau}(t)dt, \quad (11)$$

where

$$\Psi_{s,\tau}(t) = \frac{1}{\sqrt{s}}\Psi\left(\frac{t-\tau}{s}\right), \quad (12)$$

$\Psi(t)$ is a real-valued wavelet, $s = 2^j$ is the scale and $\tau = 2^j k$ is the position value (both based on power of two).

The choice of wavelet depends upon the type of signal to be analyzed and the application. Approximation coefficient and detail coefficients obtained in result of DWT generate the matrix $\mathbf{X}$ with r columns and s rows (Fig. 3), where r is the total number of DWT coefficients; s is the number of considered responses. Matrix $\mathbf{X}$ represents the dataset with characteristics for conditions of fault-free and faulty circuit and is ready for a machine learning for the purpose of testing and fault diagnostics.

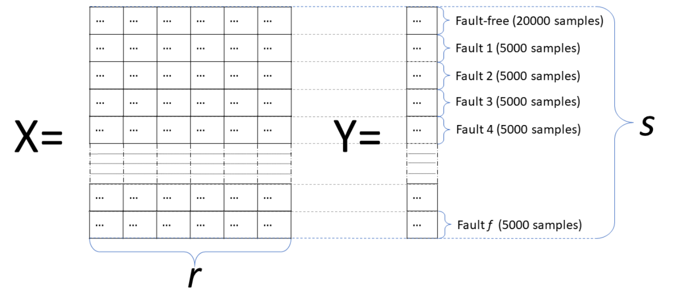

Fig. 3. The dataset structure

B. Class Aggregation and Building Model

This technique is used to aggregate the class labels of different kinds of faults. It is required to increase the completeness and accuracy of the machine learning model. As mentioned in Section II different circuit's conditions may partially or completely intersect as well as consist of values, which do not intersect with values from other sets. This technique of class aggregation is often used in classification tasks with unbalanced samples. The main task of quality checking is not to miss any fault in the circuit. Therefore, fault coverage is always a priority. Class aggregation step consists of two parts.

In the first part, the predictions based on *k*-nearest neighbors algorithm (*k-NN*) are realized. Dataset from wavelet-transformation is splitted into the training and testing subsets. A model based on *k-NN* which is non-parametric method used for classification and regression is built on the train part. The algorithm is able to distinguish among all the observations of the *k* known objects similar to new previously unknown object

which is based on the classes of the nearest neighbors of the decision regarding the new object. Despite its simplicity, *k-NN* can outperform more powerful classifiers and is used in a variety of applications such as economic forecasting, data compression and genetics [5]–[7].

In the second part, the association rule extraction algorithm is run on the prediction from *k-NN*. Association rule problem statement is well-defined [8], [9].

The following definition is used to process the association rules algorithm:

Definition 1: Let $I = \{i_1, i_2, i_3 ... i_d\}$ be a set of different attributes and the database $D = \{t_1, t_2, t_3 .... t_N\}$ where $t_N \subseteq I$. The association rule is an implication of the form of $X \implies Y$, where $X, Y \subseteq I$ and $X \cap Y = \emptyset$. Measures of significance are used to select interesting rules from the set of all possible rules

$$supp(X) = \frac{|\{t \in T; X \subseteq t\}|}{|T|}, \qquad (13)$$

where the support (called $supp$) is the ratio of the number of transactions containing $X \cup Y$. The problem of mining association rules is in the generation of rules consider the user specified minimum support (called $minsup$). According to the Eq. 14, the algorithm generates $3^d + 2^{d+1} + 1$ rules, where d is a number of unique classes. The minimum support threshold value solely depends on the database, and its optimal value is not possible to know in advance. However, excessively high values will result in the loss of values in the interesting rare classes. Very low value will increase the computational cost in the calculation of large itemsets.

$$R = \sum_{d-1}^{k=1} \left[\binom{d}{k} \times \sum_{d-k}^{j=1} \binom{d-k}{j} \right] = 3^d + 2^{d+1} + 1. \qquad (14)$$

Prediction for the test part is used for association rule extraction to decrease the number of classes. The data-flow of the algorithm's work is represented on Figure 4. Currently there are several well-known algorithms such as Apriori, Eclat and FP-Growth [10]–[12].

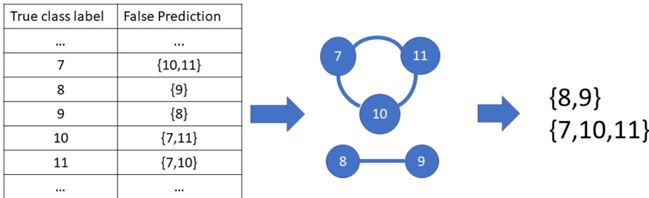

Fig. 4. An example of how an class aggregation algorithm works

The Algorithm 1 describes the pseudo-code of proposed class aggregation algorithm. This algorithm aggregate the classes based on predictions from *k-NN*.

Because of the *k-NN* algorithm nature it is highly parallelizable and well scaled with the many-core computing architectures. $Dataset(X, y)$ with minimum support threshold

Algorithm 1: Class aggregation algorithm

Data: $Dataset(X, y)$, minSupp
Result: y_{new}

1: $train_{X,y}, test_{X,y}$ = TrainTestSplit(Dataset) ;
2: model = KNeighborsClassifier($train_{X,y}$) ;
3: $y_{prediction}$ = model.prediction($test_X$) ;
4: L_1 = findFrequentClassLabels($y_{prediction}$) ;
5: **while** $L_{k-1} \neq \emptyset$ **do**
6: C_k = candidates generated from L_{k-1};
7: x = GetItemMinSupp(C_k, L_1);
8: Tgt = getTransactionID(x);
9: **foreach** $t \in D$ **do**
10: S= subset(C_k,t) ;
11: **foreach** $c \in S$ **do**
12: c.count++ ;
13: **end**
14: **end**
15: L_k = items in $C_k \geq minSupp$;
16: **end**
17: y_{new} = NewClassLabels(L_k);

value was input where the dataset is split into the train and test parts. However, each fault sample is divided individually in the ratio of 80% test and 20% train. The data was obtained from the simulation based on the Monte-Carlo method. Randomness is used in the method of Monte-Carlo, which is why the test and the training samples are not chosen randomly from the dataset. However, in each split, the proportion of faults is preserved, as in the input data set. The first model was trained and predicted test part on line 3 and 4.

The first pass of the Apriori algorithm counts the item occurrences to determine the 1–itemsets of faults. Apriori algorithm part prunes those candidates of faults combinations for which a subset is known to be infrequent. Usage of the minimum support threshold significantly reduces the search space of itemsets. The iterations begin with size 1-itemsets, and the size is incremented after each iteration. The algorithm terminates when no further successful extensions are found which means if C_k is empty for some k. Based on Apriori algorithm, new fault number was assigned on line 17 and returned new vector y_{new} as the result.

The precision, fault coverage (recall) and F_1 score (15) are used for the evaluation of classification model [13]. These metrics are widely used measures to evaluate a classification model.

$$F_1 = \frac{2 \times precision \times faultcoverage}{precision + faultcoverage} \qquad (15)$$

Fault coverage is defined as the ratio of correct assignments by the system divided by the total number of correct assignments. Precision is the ratio of correct assignments by the system divided by the total number of the system's assignments.

IV. Experimental results

Active filters are the most commonly used elements for radio engineering and especially important for audio equipment, signal processing systems, measuring instruments.

The bandpass Sallen-Key filter was used for experiments (Figure 5). Sallen-Key filters are very convenient in batch production since they require parts of the same denominations and with a large allowable deviation. Implemented as a simple circuit with two resistors, two capacitors and an operational amplifier, representing a filter with the second order transfer function. Filters of higher order can be obtained by the connection of the elementary filters in series. The filter can have an arbitrary gain bandwidth. The experimental filter has the following nominals: $R_1 = 10k, R_2 = 20k, R_3 = 10k, R_a = 5k, R_b = 10k, C_1 = 220n, C_2 = 220n$.

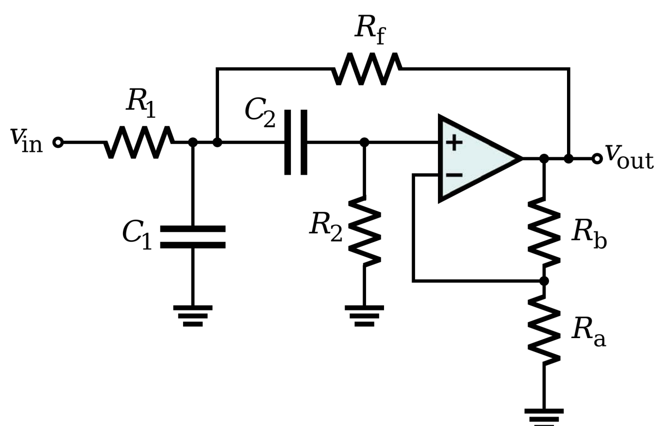

Fig. 5. Sallen-Key bandpass filter

The sine wave with frequency 72 Hz and amplitude $1V$ was used as a test signal applied to the input of the filter. The steady-state output responses V_{out} are measured at the test node for a period. Wavelet transformation using the Daubechies wavelet of fourth order (db4) is carried out to the each output response. As a result, each response is represented by 148 coefficients.

The fault list includes 28 faults and one fault-free circuit condition. In turn to, the 28 faults are represented by two types of catastrophic and parametric faults. Catastrophic faults included short circuit and open circuit effects for each component. Parametric faults included deviations of component parameters on + 50% and – 50 % from the nominal. The Monte-Carlo simulation used 20 000 iterations for fault-free condition and 5 000 for each fault. The particular number of the circuit condition (fault-free and faulty) is included into associated vector Y for each simulation accordingly (Fig. 3). Resulting data structure is also presented the table, where the rows are the number of simulations, and the column is the number of co-factors after the transformation. Also, the structure of the data was ready for the application of machine learning algorithms.

The *k-NN* algorithm was trained on the training part. The model's prediction on test part automatically occurred after training. Association algorithm based on prediction generates the new vector Y_{new}, where overlapping faults labels merged together. According to Eq. 14 the total number of association rules is 6.86×10^{13}, because an experimental number of unique faults is 28 and one fault-free condition.

The results are shown in Table 1. The initial *k-NN* model was retrained based on the new vector of Y_{new}. New generated *k-NN* model provided the following result as presented in the Table 2.

According to the results, a reasonable accurate model was obtained which was also had a wide coverage of all faults.

The experiment was done on system with Intel®Core™i5-4590 3.30 GHz, 16 GB of RAM and Windows 10 Pro. Python programming language version 3.6 was used for the technique implementation.

V. Conclusion

The technique of construction for the classifier for analog fault testing and diagnosis was done by using the extraction of the essential characteristics based on wavelet transformation, Monte-Carlo method, association rules mining algorithms, and machine learning algorithm. The proposed technique helps to produce the high reliable analog and mixed-signals integrated circuits. The experimental verification of the prediction quality was performed on the most widely used filter topologies. The results obtained for the Sallen-Key filter demonstrate the high precision of prediction ($> 99,09\%$) and fault coverage ($> 99,08\%$) in the task of fault diagnostics. The proposed technique uses algorithms which were parallel and prepared to handle the big data obtained in result of the exhaustive simulation of analog circuits.

VI. Acknowledgment

We would like to express great appreciation to Professor Joshua Zhexue Huang for his valuable and constructive suggestions during the planning and development of this research work. The work is performed according to the Russian Government Program of Competitive Growth of Kazan Federal University.

References

[1] S. Mosin, "Design-for-testability automation of mixed-signal integrated circuits," in *2013 IEEE International SOC Conference*. IEEE, Sept 2013, pp. 244–249. [Online]. Available: http://ieeexplore.ieee.org/document/6749695/

[2] ——, "Automated simulation of faults in analog circuits based on parallel paradigm," in *2017 IEEE East-West Design Test Symposium (EWDTS)*. IEEE, Sept 2017, pp. 1–6. [Online]. Available: http://ieeexplore.ieee.org/document/8110133/

[3] H. G. Stratigopoulos and S. Sunter, "Efficient Monte Carlo-based analog parametric fault modelling," in *Proceedings of the IEEE VLSI Test Symposium*. IEEE, apr 2014, pp. 1–6. [Online]. Available: http://ieeexplore.ieee.org/document/6818741/

[4] A. Kavithamani, V. Manikandan, and N. Devarajan, "Soft fault classification of analog circuits using network parameters and neural networks," in *Journal of Electronic Testing: Theory and Applications (JETTA)*, vol. 29, no. 2. Springer US, apr 2013, pp. 237–240. [Online]. Available: http://link.springer.com/10.1007/s10836-013-5370-3

TABLE I
PRESENTED CLASSIFICATION REPORT OF *k-NN* WITHOUT CLASS AGGREGATION ALGORITHM

fault label	precision	fault coverage	F_1-score
0	0.7824	0.9526	0.8591
1	0.0357	0.0062	0.0105
2	0.9969	1.0000	0.9985
3	1.0000	1.0000	1.0000
4	0.8742	1.0000	0.9329
5	0.9990	1.0000	0.9995
6	1.0000	1.0000	1.0000
7	0.3879	0.5439	0.4528
8	0.8738	0.8232	0.8477
9	0.8381	0.8850	0.8609
10	0.2054	0.2111	0.2082
11	0.1274	0.0719	0.0920
12	1.0000	0.9613	0.9803
13	1.0000	0.8573	0.9232
14	1.0000	1.0000	1.0000
15	0.9894	0.9442	0.9662
16	0.9829	0.9942	0.9885
17	0.7783	0.7618	0.7700
18	0.9719	0.9603	0.9661
19	0.8974	0.8887	0.8930
20	0.9737	0.9708	0.9723
21	0.8851	0.8940	0.8896
22	0.9772	0.9743	0.9757
23	0.9886	0.9905	0.9896
24	1.0000	0.9980	0.9990
25	0.9951	0.9990	0.9971
26	0.9899	0.9980	0.9939
27	0.7586	0.7729	0.7657
28	0.9371	0.9715	0.9540
avg/total	0.8310	0.8521	0.8392

TABLE II
CLASSIFICATION REPORT OF *k-NN* WITH CLASS AGGREGATION ALGORITHM

fault label	precision	fault coverage	F_1-score
{0,1}	0.9979	0.9994	0.9987
2	0.9939	1.0000	0.9969
3	1.0000	1.0000	1.0000
4	1.0000	1.0000	1.0000
5	0.9971	1.0000	0.9985
6	1.0000	1.0000	1.0000
{7,10,11}	1.0000	1.0000	1.0000
{8,9}	1.0000	1.0000	1.0000
12	1.0000	0.9516	0.9752
13	1.0000	1.0000	1.0000
14	1.0000	1.0000	1.0000
15	1.0000	0.9939	0.9969
16	0.9981	0.9971	0.9976
17	0.9419	0.9383	0.9401
18	0.9870	0.9802	0.9836
19	0.9832	0.9736	0.9784
20	1.0000	1.0000	1.0000
21	0.9670	0.9768	0.9719
22	0.9920	0.9881	0.9901
23	0.9896	0.9914	0.9905
24	0.9970	0.9961	0.9966
25	0.9990	1.0000	0.9995
26	0.9647	1.0000	0.9820
27	0.9347	0.9402	0.9374
28	0.9748	0.9867	0.9807
avg/total	0.9909	0.9908	0.9908

[5] M. Quispe-Ayala, K. Asalde-Alvarez, and A. Roman-Gonzalez, “Image classification using data compression techniques,” *Engineers in Israel*, pp. 349–353, nov 2010.

[6] N. K. Ahmed, A. F. Atiya, N. E. Gayar, and H. El-Shishiny, “An Empirical Comparison of Machine Learning Models for Time Series Forecasting,” *Econometric Reviews*, vol. 29, no. 5-6, pp. 594–621, aug 2010.

[7] A. H. Asikainen, J. Ruuskanen, and K. A. Tuppurainen, “Consensus kNN QSAR: A versatile method for predicting the estrogenic activity of organic compounds in silico. A comparative study with five estrogen receptors and a large, diverse set of ligands,” *Environmental Science and Technology*, vol. 38, no. 24, pp. 6724–6729, nov 2004. [Online]. Available: http://pubs.acs.org/doi/abs/10.1021/es049665h http://ieeexplore.ieee.org/document/5662206/

[8] R. Agrawal, T. Imielinski, and A. Swami, “Mining association rules between standets of items in large databases,” *Proceedings of the 1993 ACM SIGMOD international conference on Management of data*, vol. 22, no. May, pp. 207–216, 1993.

[9] C. Ordonez and E. Omiecinski, “Discovering association rules based on image content,” in *Proceedings IEEE Forum on Research and Technology Advances in Digital Libraries*. IEEE Comput. Soc, 1999, pp. 38–49. [Online]. Available: http://ieeexplore.ieee.org/document/777689/

[10] R. Agrawal and R. Srikant, “Fast algorithms for mining association rules,” in *94 Proceedings of the 20th International Conference on Very Large Data Bases*, vol. 1215, 1994, pp. 487–499.

[11] M. J. Zaki and M. J. Zaki, “Scalable Algorithms for Association Mining,” *IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING*, vol. 12, pp. 372—-390, 2000. [Online]. Available: http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.28.656

[12] J. Han, J. Pei, and Y. Yin, “Frequent Pattern Tree : Design and Construction,” *Networks*, vol. 29, no. 2, pp. 1–12, 2000. [Online]. Available: http://portal.acm.org/citation.cfm?doid=342009.335372

[13] C. J. Van Rijsbergen, *Information Retrieval*. Butterworths, 1979, vol. 573. [Online]. Available: http://link.springer.com/10.1007/978-3-319-41718-9