Regular Paper
J. Korean Inst. Electr. Electron. Mater. Eng.
Vol. 31, No. 7, pp. 450-454 November 2018
DOI: https://doi.org/10.4313/JKEM.2018.31.7.450
ISSN 1226-7945(Print), 2288-3258(Online)

기계학습 알고리즘을 이용한 반도체 테스트공정의 불량 예측

장수열, 조만식, 조슬기, 문병무[a]
고려대학교 전기전자공학과

Defect Prediction Using Machine Learning Algorithm in Semiconductor Test Process

Suyeol Jang, Mansik Jo, Seulki Cho, and Byungmoo Moon[a]
Department of Electrical Engineering, Korea University, Seoul 02841, Korea

(Received August 21, 2018; Revised September 8, 2018; Accepted September 11, 2017)

Abstract: Because of the rapidly changing environment and high uncertainties, the semiconductor industry is in need of appropriate forecasting technology. In particular, both the cost and time in the test process are increasing because the process becomes complicated and there are more factors to consider. In this paper, we propose a prediction model that predicts a final “good” or “bad” on the basis of preconditioning test data generated in the semiconductor test process. The proposed prediction model solves the classification and regression problems that are often dealt with in the semiconductor process and constructs a reliable prediction model. We also implemented a prediction model through various machine learning algorithms. We compared the performance of the prediction models constructed through each algorithm. Actual data of the semiconductor test process was used for accurate prediction model construction and effective test verification.

Keywords: Machine learning, Semiconductor test process, Prediction model, Classification, Package test

1. 서 론

반도체 산업은 스마트폰과 태블릿 등 모바일 기기에서부터 자동차와 로봇 분야에 이르기까지 여러 산업의 큰 영향을 미치며 빠르게 성장해 나가고 있다. 반도체 제조 공정은 전자산업에서 가장 복잡하고 정밀성이 요구되는 공정으로, 빠른 기술의 변화와 높은 불확실성으로 인하여 적절한 과학기술적 방법론과 생산전략이 필수적이다. 반도체 공정의 흐름은 대표적으로 wafer fabrication (Fab 공정), 프로브 검사(probe test), 조립 공정(assembly) 그리고 패키지 검사(Package test)로 구성되어 있다 [1]. Fab 공정을 거치면서 한 장의 웨이퍼 위에 수천 개의 집적회로(integrated circuit, IC) 칩(chip)이 생성된다. 프로브 검사는 웨이퍼에 생성된 ic칩이 정상적인 전기적 특성을 보이는지 검사하여 양/불량을 판별하는 단계이다. 조립 공정은 만들어진 칩들을 분리하여 전기적·물리적 특성을 향상시키고 외부의 물리적 충격으로부터 칩을 보호하기 위한 공정이다. 패키지 검사는 조립된 칩들의 전기적 특성과 신뢰성을 검사하여 최종적으로 양품과 불량품을 구분하는 단계이다. 완성된 반도체 칩들은 preconditioning test을 포함하여, TCT (thermal cycling test), THT (temperature/humidity test), HAST (highly accelerated stress test) 등과 같은 다양한 신뢰성 시험들을 거치게 된다.

Preconditioning test는 패키지의 수분 흡습에 따른 불량 발생 정도를 평가하기 위해 수행되는 테스트로서, 테스트 과정에서 수분과 온도, 전기적 특성 등 다양한 계측인자가 포함된다 [2]. 그림 1은 preconditioning test의 과정을 보여준다.

a. Corresponding author; byungmoo@korea.ac.kr

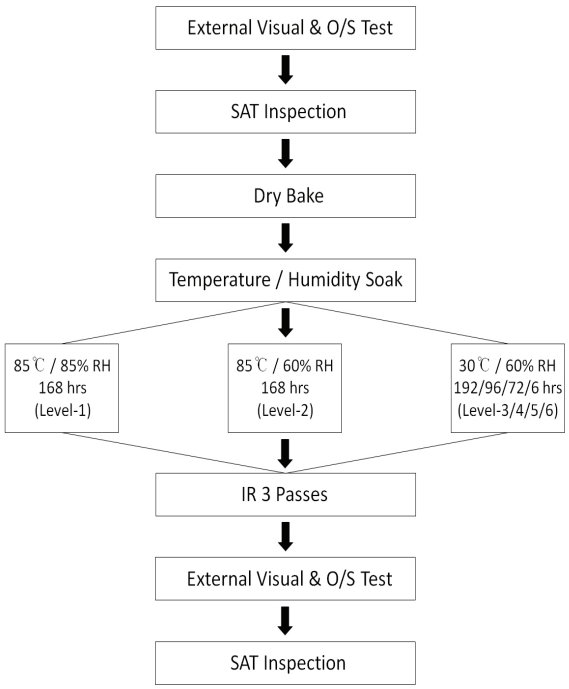

Fig. 1. Preconditioning test procedure.

반도체 공정이 복잡해지고 고려할 변수가 많아지면서, 테스트 공정의 시간과 비용은 계속해서 증가하고 있다. Preconditioning test를 통과한 칩이 다른 신뢰성 검사들을 통과하는지 미리 예측할 수 있다면, 불량 칩에 대하여 수행되는 테스트의 비용과 시간을 절감할 수 있기 때문에 매우 효율적일 것이다.

본 연구에서는 반도체 테스트 공정에서 실제로 계측된 preconditioning test의 데이터를 바탕으로 최종적인 양/불량 여부를 예측하는 예측 모델을 구현하고자 한다. Preconditioning test를 통과한 칩의 테스트 측정값을 입력 인자로 하여 최종검사에서의 양/불량 여부를 예측하여 분류하는 학습 모델을 구축하는 것이다. 이를 통해 조기에 불량 제품을 선별 폐기하고 신속하게 양품으로 전환 생산하여 반도체 제조 생산성 향상 및 비용 절감 효과를 기대할 수 있다.

2. 실험 방법

2.1 데이터 전처리

그림 2에서 보여주듯이 예측 모델링의 첫 번째 단계는 데이터 전처리 작업이다. 테스트 과정에서 계측된

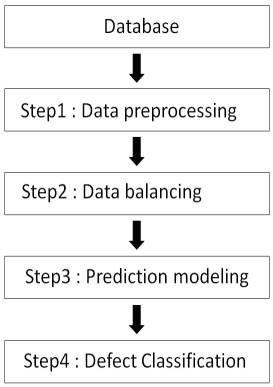

Fig. 2. Overview of the prediction modeling procedure.

수십 가지의 변수들 중 종속변수에 영향을 주는 독립변수들을 선별하였다. 동시에, 종속변수와 관련이 없는 변수들은 모델링의 효율성과 정확성을 위하여 제거하였다. 변수 선택의 기법은 상관관계 분석(correlation analysis), 카이스퀘어 검정(chi-square test) 등의 통계적 분석과 반도체 지식을 바탕으로 수행되었다.

변수 선택 외에도 결측 값(missing value)과 이상 값(outlier)을 제거하는 작업을 수행하였다. 계측되는 데이터의 수가 대단히 많고, 장비의 오작동이 발생하기 때문에 데이터에는 대부분 결측 값과 이상 값이 포함된다. 결측 값과 이상 값 역시 정확한 모델을 구축하는 데 있어 방해 요인으로 작용하므로 제거하였다. 본 연구에서는 비지도 이상치 탐지 기법(unsupervised anomaly detection)을 사용하여 이상 값을 제거하였다 [3].

2.2 데이터 불균형 해결

현실에서 마주하게 되는 데이터의 상당수는 불균형 상태에 놓여 있고 [4], 이것은 알고리즘의 예측 성능을 저하시키는 주된 요인 중 하나이다 [5]. 칩의 양/불량을 구분하는 모델을 구축하기 위해서는 데이터의 불균형 문제를 해결해야 한다. 본 연구에서는 SMOTE (synthetic minority over-sampling technique) 기법을 활용해 데이터 불균형 문제를 해결하였다. SMOTE 기법은 다수 계층의 과소추출과 소수 계층의 과다추출의 조합으로 데이터 불균형 문제를 해결하는 방법이다. 이 기법은 샘플링의 방법으로 kNN 기법을 사용하여 새로운 소수 계층의 데이터를 생성하는 방식으로, 소수 계층의 결정 경계를 넓히는 효과를 통해 기존의 복원 샘플링 기법이 지니고 있던 데이터 과적합(overfitting) 문제를 해결할 수 있다 [6].

2.3 예측 모델 구축

가공한 데이터를 바탕으로 예측 모델을 구축하는 단계로, 변수로 선택된 특성(feature)들이 독립변수를 구성하고, 여러 가지 기준에 의하여 양품과 불량품으로 구분된 범주형 변수를 종속변수로 활용하여 분류모델을 구축하였다. 본 연구에서는 분류 문제를 다루는 다양한 분류(classification) 알고리즘을 활용하여 알고리즘별 성능을 비교 분석해 보았다.

학습에 소요되는 시간이 짧고 결과 해석이 용이한 의사결정나무(decision tree) [7], 인스턴스 기반 학습의 가장 간단한 알고리즘인 kNN [8], 선형 모델로 분류 문제를 해결하는 로지스틱 회귀분석, 패턴 인식이나 딥 러닝(deep learning) 등 다양한 분야에 응용되는 인공신경망 [9], 최적의 초평면으로 분류 문제를 해결하는 SVM 기법 [10], 이와 함께 랜덤 포레스트(이하 RF) 기법을 사용하여 예측 모델을 구축하였다. RF 알고리즘은 분류, 회귀 분석 등에 사용되는 앙상블 학습 방법의 일종으로, 다른 분류 알고리즘에 비해 월등히 높은 정확성을 가지고 있다 [11]. 또한 모델 구축 시 결정해야 하는 매개변수의 수가 적고, 그 값들에 덜 민감하기 때문에 모델 구축이 용이하다 [12].

본 연구에서는 R의 기계학습 알고리즘을 사용하여 양/불량 예측 모델을 구축하였고, 각 알고리즘에 사용된 R 패키지와 최적의 매개변수는 표 1과 같다.

Table 1. R packages and optimal parameters of the prediction models.

Algorithm	Package	Optimal parameters
D-tree	C50	model : tree, trials : 20
kNN	class	k value : 3
Logistic	stats	family : binomial, link : logit
ANN	nnet	hidden layer : 2
SVM	Ksvm	kernel : rbfdot, cost : 1
RF	randomForest	ntree : 500, mtry : 4

2.4 양/불량 예측

Step 1에서 step 3까지의 단계적인 과정을 거치면서 구축된 분류 모델은 새로운 칩의 테스트 데이터를 바탕으로 양/불량을 예측하게 된다. 새로운 칩의 preconditioning test 데이터를 구축된 모델에 넣게 되면, 해당 칩이 최종적으로 양품일지 불량품일지 예측할 수 있는 것이다.

3. 결과 및 고찰

3.1 실험 데이터

본 연구에서는 양/불량 예측 모델의 효용성을 입증하기 위해 반도체 테스트 공정의 실제 데이터를 사용하였다. 실험에 사용된 데이터는 반도체 칩의 테스트 데이터로, 총 2,892개의 데이터로 구성되었다. 데이터는 연속형의 독립변수와 범주형의 종속변수로 이루어져 있고, 정상으로 분류된 칩 2,815개, 불량으로 분류된 칩 77개로 구성되어 있다. 데이터의 양/불량 비율은 9 대 1로 범주 간 매우 심한 불균형을 이루고 있다. 실험 결과의 안정성 확보를 위하여 10-fold cross validation으로 데이터를 처리하고 실험을 진행하였다 [13]. 이렇게 처리된 데이터의 구성은 표 2에서 확인할 수 있다. 또한, 효과적이고 합리적인 테스트를 위해 데이터에 대한 정규화 작업을 진행하였다. 변수마다 측정 인자와 범위, 단위 등이 다 다르기 때문에 정확한 결과값을 얻기 위해서는 데이터의 정규화 작업이 필수적이다. 정규화의 방법으로는 z-score를 사용하였다.

Table 2. 10-fold cross validation data for an experiment.

Data	Class	Fold 1	Fold 2	Fold 3	Fold 4	Fold 5	Fold 6	Fold 7	Fold 8	Fold 9	Fold 10
Train	Good	2,339	2,339	2,337	2,338	2,334	2,341	2,338	2,346	2,338	2,337
	Bad	263	263	265	265	259	262	265	267	267	266
Test	Good	259	259	260	259	256	258	259	261	261	260
	Bad	31	31	30	30	33	31	30	28	28	29

3.2 실험 성능 평가

분류문제에 일반적으로 사용되는 성능 척도는 정확도(accuracy)로, 그림 3에서 보여주는 혼동행렬을 통해 구할 수 있다. 하지만 범주 간 불균형이 심한 데이터의 경우는 대부분의 관측치가 특정 범주에 편향되어 분포하기 때문에, 기존의 정확도를 성능 평가 척도로 사용하기 어렵다. 이 경우는 거의 모든 데이터를 다수 범주로 분류하게 되고, 90% 이상의 높은 정확도를 얻게 된다. 하지만 소수 범주에 속한 데이터를 거의 판별하지 못하는 커다란 문제점을 지니게 된다.

따라서 본 연구에서는 성능 평가 요소로 기존의 정확도를 대신하여, 불균형 데이터 분류 문제에서 평가척도로 사용되는 민감도(sensitivity)와 특이도(specificity)의 기하평균(이하 GM)을 사용하였다 [14]. GM은 식 (1)과 같다.

$$GM = \sqrt{Sensitivity \times Specificity} \tag{1}$$

민감도는 실제 불량으로 분류된 칩 대비 모델이 올바르게 불량으로 예측한 경우의 비율을 의미하고 식 (2)와 같이 구할 수 있다. 특이도는 실제 정상으로 분류된 칩 대비 모델이 올바르게 정상이라고 예측한 경우의 비율을 의미하고 식 (3)과 같이 구할 수 있다. GM은 민감도와 특이도의 곱의 제곱근으로 계산되어지며, 불균형 데이터의 분류 문제에서 기존의 정확도가 가진 한계를 뛰어넘을 수 있는 성능 평가 방법이다.

		Predict	
		Positive	Negative
Actual	Positive	True Positive (TP)	False Negative (FN)
	Negative	False Positive (FP)	True Negative (TN)

Note) TP : the proportion of low yielde that are correctly predicted, FN : the proportion of low yields that are incorrectly predicted as high yields, TN : the proportion of high yields that are correctly predicted, FP : the proportion of high yields that are incorrectely predicted as low yields.

Fig. 3. Confusion matrix.

$$Sensitivity = \frac{TP}{(TP+FN)} \tag{2}$$

$$Specificity = \frac{TN}{(TN+FP)} \tag{3}$$

3.3 실험 결과

표 3은 RF 알고리즘을 이용하여 모델을 구축했을 때, 데이터의 불균형 정도에 따른 성능의 차이를 나타낸 것이다. SMOTE(100:100)은 소수계층의 과다추출과 다수계층의 과소추출 비율이 동일한 샘플링 환경을 의미한다.

SMOTE(200:100)부터 SMOTE(500:100)은 소수계층의 과다추출 샘플링 개수를 점진적으로 증가시킨 것이다. 불량 칩 데이터의 샘플링을 과도하게 수행할수록 예측 정확도가 낮아지는 것을 확인 할 수 있었다. 또한, SMOTE 기법을 적용하지 않고 구축한 모델의 경우는 GM이 72%로, SMOTE 기법을 적용한 모델과 비교하여 GM이 상당히 떨어짐을 알 수 있었다. 양/불량의 비율을 동일하게 하여 샘플링한 SMOTE(100:100)의 경우가 가장 높은 성능을 보였다. 표 4는 본 연구에서 제안한 양/불량 예측 방법을 다양한 기계학습 알고리즘으로 적용하여 모델 간 성능을 비교한 것이다. 가장 높은 성능을 보였던 SMOTE(100:100)의 경우는 알고리즘별로 큰 성능의 차이를 보이지 않았다. 하지만 SMOTE를 적용하지 않은 경우는 알고리즘에 따라 최대 47%의 큰 성능 차이를 보였다. RF 알고리즘을 사용하여 구축한 예측 모델이 가장 우수한 성능을 보였고, 인경 신경망 알고리즘을 사용한 예측 모델이 비슷한 성능으로 그 뒤를 이었다.

주목할 점은 인공 신경망 모델이 민감도 기준 98%의 예측 정확도를 나타냈다는 것이다. 이것은 인공 신

Table 3. Performance of RF for different degrees of imbalance.

Dataset	Sensitivity	Specificity	GM
SMOTE (100:100)	95%	92%	93%
SMOTE (200:100)	81%	93%	90%
SMOTE (300:100)	69%	95%	81%
SMOTE (400:100)	66%	97%	79%
SMOTE (500:100)	53%	99%	72%
Without using SMOTE	23%	99%	47%

Table 4. Experiment result (for all of the methods).

Dataset	Model	Sensitivity	Specificity	GM
SMOTE (100:100)	D-tree	93%	87%	90%
	kNN	96%	82%	89%
	Logistic	95%	87%	91%
	ANN	98%	87%	92%
	SVM	95%	87%	91%
	RF	95%	92%	93%
Without using SMOTE	D-tree	56%	98%	74%
	kNN	19%	99%	43%
	Logistic	31%	99%	56%
	ANN	38%	99%	61%
	SVM	8%	99%	27%
	RF	23%	99%	47%

경망 알고리즘 모델이 소수 범주 분류 문제에서 우수한 예측 성능을 보일 수 있다는 것이다. 또한, SMOTE를 적용하지 않은 모델에서, 의사결정나무 알고리즘을 사용하여 구축한 모델이 다른 알고리즘들과 비교하여 민감도를 기준으로 높은 예측 성능을 보였다. 이는 곧, 의사결정나무 알고리즘은 데이터가 불균형한 경우에도 어느 정도 예측 성능을 낼 수 있음을 의미한다. 반대로 SVM 알고리즘은 데이터가 불균형할 경우 매우 낮은 예측 성능을 보였다. SVM 알고리즘은 SMOTE의 적용 여부에 따라 매우 큰 예측 성능의 차이를 보였다.

4. 결 론

본 논문에서는 반도체 테스트 공정에서 생성된 데이터를 바탕으로 반도체 칩의 양/불량을 판별할 수 있는 예측 모델을 제안하였다. 예측 모델은 반도체 테스트 데이터를 바탕으로 데이터 전처리 과정과 데이터의 불균형 문제를 해결하는 과정을 거쳐, 기계학습 분류 알고리즘을 통하여 구축되었다. 구축된 예측 모델은 preconditioning test 데이터를 통해 최종적인 반도체 칩의 양/불량 여부를 예측하게 된다. 이러한 예측 모델을 이용하면, 추가적으로 진행되는 테스트에 소모되는 시간과 비용 모두를 절약할 수 있을 것이다. 또한, 변수별 분석이나 상관분석이 가능해지므로 제조 공정 능력 향상에 기여할 수 있을 것이다.

실험 결과, SMOTE 기법을 사용한 dataset이 일반 dataset보다 예측 성능이 높았다. 알고리즘별 비교에서는 RF 알고리즘이 가장 뛰어난 성능을 보였다. SMOTE 기법 미적용의 불균형 dataset에서는 decision tree 알고리즘이 74%로 가장 우수한 성능을 보였고, SVM 알고리즘이 27%의 가장 저조한 성능을 보였다. SMOTE 기법을 사용한 경우는 알고리즘별 최대 4% 내외의 성능 차이를 보였지만, SMOTE 미적용의 경우는 최대 47%의 성능 차이를 보였다.

REFERENCES

[1] R. Uzsoy, C. Y. Lee, and L. A. Martin-Vega, *J. IIE Trans.*, **24**, 47 (1992). [DOI: https://doi.org/10.1080/07408179208964233]

[2] X. Fan, G. Q. Zhang, W. D. van Driel, L. J. Ernst, *IEEE Trans. Compon. Packag. Technol.*, **31**, 252 (2008). [DOI: https://doi.org/10.1109/TCAPT.2008.921629]

[3] V. Chandola, A. Banerjee, and V. Kumar, *J. ACM Comput. Surv.*, **41**, 15 (2009). [DOI: https://doi.org/10.1145/1541880.1541882]

[4] C. Chen, A. Liaw, and L. Breiman, *University of California, Berkeley*, **110**, 1 (2004).

[5] P. Kang and S. Cho, *Int. Conf. Neural Inf. Process.*, **4232**, 837 (2006). [DOI: https://doi.org/10.1007/11893028_93]

[6] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, *J. Artif. Intell. Res.*, **16**, 321 (2002). [DOI: https://doi.org/10.1613/jair.953]

[7] J. R. Quinlan, *Mach. Learn.*, **1**, 81 (1986). [DOI: https://doi.org/10.1007/BF00116251]

[8] N. S. Altman, *Am. Stat.*, **46**, 175 (1992). [DOI: https://doi.org/10.1080/00031305.1992.10475879]

[9] F. Rosenblatt, *Psychol. Rev.*, **65**, 386 (1958). [DOI: https://doi.org/10.1037/h0042519]

[10] V. N. Vapnik, *IEEE Trans. Neural Networks*, **10**, 988 (1999). [DOI: https://doi.org/10.1109/72.788640]

[11] L. Breiman, *Mach. Learn.*, **45**, 5 (2001). [DOI: https://doi.org/10.1023/A:1010933404324]

[12] A. Liaw and M. Wiener, *R News*, **2**, 18 (2002).

[13] R. Kohavi, *Proc. IJCAI '95 Proceedings of the 14th International Joint Conference on Artificial Intelligence - Volume 2* (Montreal, Quebec, Canada, 1995) p. 1137.

[14] M. Kubat, R. C. Holte, and S. Matwin, *Mach. Learn.*, **30**, 195 (1998). [DOI: https://doi.org/10.1023/A:1007452223027]