반도체공정 이상탐지 및 클러스터링을 위한 심볼릭 표현법의 적용

(Application of Symbolic Representation Method for Fault Detection and Clustering in Semiconductor Fabrication Processes)

노 웅 기 † 홍 상 진 ††
(Woong-Kee Loh) (Sang Jeen Hong)

요 약 반도체(semiconductor) 기술은 1950년대에 집적 회로(integrated circuit, IC)가 발명된 이후 오늘날까지 급속한 발전을 거듭하고 있다. 하나의 완전한 반도체를 제조하기 위해서는 매우 다양하고 긴 공정을 거쳐야 한다. 반도체 제조 생산성을 높이기 위하여 공정들이 종료되기 전에 미리 이상(fault)을 발견하기 위한 이상탐지 및 분류(fault detection and classification, FDC)에 대한 많은 연구가 진행되고 있다. 이를 위하여 다양한 반도체 장비에 갖가지 종류의 센서를 부착하여 일정한 시간 간격으로 원하는 값을 측정한다. 이러한 측정 값은 실수 값들의 연속이므로 시계열(time-series) 데이터의 일종이다. 본 논문에서는 반도체 공정에서의 이상탐지 및 클러스터링을 수행하는 알고리즘을 제안한다. 제안된 알고리즘은 시계열 데이터를 심볼릭 표현법(symbolic representation)으로 변환하여 이상을 탐지하는 기존의 알고리즘을 수정한 것이다. 본 논문의 공헌은 일반적인 시계열 데이터에 대한 기존의 이상탐지 알고리즘을 수정하여 반도체 공정 데이터에 대해서도 활용할 수 있음을 보일 뿐만 아니라, 이상탐지 및 클러스터링의 정확성을 높이는 실험 결과를 제시하는 것이다. 실험 결과, 본 논문에서 제안한 알고리즘은 긍정 오류(false positive), 및 부정 오류(false negative)를 모두 발생하지 않았다.

키워드 : 심볼릭 표현법, 시계열 데이터, 반도체 공정, 이상탐지, 클러스터링, 알고리즘

Abstract Since the invention of the integrated circuit (IC) in 1950s, semiconductor technology has undergone dramatic development up to these days. A complete semiconductor is manufactured through a diversity of processes. For better semiconductor productivity, fault detection and classification (FDC) has been rigorously studied for finding faults even before the processes are completed. For FDC, various kinds of sensors are attached in many semiconductor manufacturing devices, and sensor values are collected in a periodic manner. The collection of sensor values consists of sequences of real numbers, and hence is regarded as a kind of time-series data. In this paper, we propose an algorithm for detecting and clustering faults in semiconductor processes. The proposed algorithm is a modification of the existing anomaly detection algorithm dealing with symbolically-represented time-series. The contributions of this paper are: (1) showing that a modification of the existing anomaly detection algorithm dealing with general time-series could be used for semiconductor process data and (2) presenting experimental results for improving correctness of fault detection and clustering. As a result of our experiment, the proposed algorithm caused neither false positive nor false negative.

Key words : symbolic representation, time-series, semiconductor process, fault detection, clustering, algorithm

· 본 논문은 2008년 지식경제부 전략기술개발사업의 지원을 받아 수행한 연구임 (10031812-2008-11)

† 정 회 원 : 성결대학교 멀티미디어학부 교수
woong@sungkyul.ac.kr
†† 정 회 원 : 명지대학교 전자공학과 교수
samhong@mju.ac.kr

논문접수 : 2009년 7월 29일
심사완료 : 2009년 8월 25일

정보과학회논문지 : 컴퓨팅의 실제 및 레터 제15권 제11호(2009.11)

1. 서 론

반도체(semiconductor) 기술은 1950년대에 집적 회로(integrated circuit, IC)가 발명된 이후 오늘날까지 급속한 발전을 거듭하고 있다. 반도체는 끊임없이 고집적화, 고성능화, 소형화, 경량화되고 있으며, 현대의 거의 모든 장치에 내장되어 있다. 이러한 추세에 따라 반도체 공정은 더욱 높은 정밀도를 요구하고 있으며, 완전한 반도체를 생산하기 위하여 공정 중에 아주 작은 오류조차 허용되지 않는다.

하나의 완전한 반도체를 제조하기 위해서는 매우 다양하고 긴 과정을 거쳐야 한다. 과거에는 이러한 각 과정을 이상(fault) 없이 완료하기 위한 공정 변이범위(process variation range)를 통계적으로 산출하였다[1,2]. 공정 변이범위란 그 공정을 완전하게 끝낼 수 있도록 공정 내에서 변화를 줄 수 있는 범위를 의미한다. 예를 들어, 특정 공정에서 이온의 에너지가 수백 eV를 유지해야 한다면, 그 공정에서의 이온의 에너지 변이범위를 100~1000eV로 설정할 수 있다. 이러한 변이범위를 설정하는 방법을 통계적 공정제어(statistical process control, SPC)라고 한다[2].

반도체를 제조하는 세부 공정들 중에는 피할 수 없거나 효율성의 이유로 기술자의 중간 관여 없이 연속적으로 수행되는 공정들이 있으며, 만약 이러한 공정들 중에 이상이 발생했다면 그러한 연속된 공정들이 모두 종료되어야만 그 이상을 발견할 수 있다. 이러한 문제를 극복하기 위하여 그러한 연속된 공정들이 종료되기 전에 미리 이상을 발견하기 위한 고급 공정제어(advanced process control, APC) 분야가 최근에 관심을 끌고 있다[1,3]. 특히, 이상탐지 및 분류(fault detection and classification, FDC)에 대한 많은 연구가 진행되고 있다. FDC를 수행하기 위하여 다양한 반도체 장비에 갖가지 종류의 센서를 부착하여 일정한 시간 간격으로 원하는 값을 측정하여 이상을 탐지한다. 이러한 측정 값은 실수 값들의 연속이므로 시계열(time-series) 데이터의 일종이다.

본 논문에서는 반도체 공정에서의 이상탐지 및 클러스터링을 수행하는 알고리즘을 제안한다. 제안된 알고리즘은 시계열 데이터를 심볼릭 표현법(symbolic representation)으로 변환하여 이상을 탐지하는 기존의 알고리즘[4]을 수정한 것이다. 심볼릭 표현법이란 연속된(continuous) 실수 값으로 구성된 시계열 데이터를 한정된 개수의 이산(discrete) 심볼로 표현하는 방법으로, 다양한 심볼릭 표현법들이 제안되었다[5,6].

본 논문의 공헌은 일반적인 시계열 데이터에 대한 기존의 이상탐지 알고리즘을 수정하여 반도체 공정 데이터에 대해서도 활용할 수 있음을 보일 뿐만 아니라, 이상탐지 및 클러스터링의 정확성을 높이는 실험 결과를 제시하는 것이다. 특히, 본 논문에서 이상탐지의 대상이 되는 반도체 공정 데이터는 실제의 공정에서 얻어진 데이터이며, 그 용량도 기존의 시계열 데이터 연구에서 사용된 것들과 비교하여 대용량이다. 실험 결과, 본 논문과 같은 데이터를 대상으로 신경망(neural network) 기법을 이용한 참고문헌[1]에서 긍정 오류(false positive)가 발생한 반면, 본 논문에서 제안한 알고리즘은 긍정 오류뿐만 아니라 부정 오류(false negative)도 발생하지 않았다.

본 논문의 구성은 다음과 같다. 제2절과 3절에서는 각각 반도체 공정과 심볼릭 표현법에 대하여 간략하게 설명한다. 제4절과 5절에서는 반도체 공정 데이터에 대한 이상탐지 및 클러스터링 알고리즘에 대하여 설명한다. 제6절에서는 제안된 알고리즘에 대한 실험 결과를 보이고, 제7절에서 결론을 맺는다.

2. 반도체 공정

본 절에서는 본 논문에서 제안하는 이상탐지 및 클러스터링 알고리즘의 대상 데이터인 반도체 공정 데이터에 대한 이해를 돕고자 Complementary Metal-Oxide-Semiconductor(CMOS)를 제작하는 웨이퍼(wafer) 공정의 일부를 간략하게 설명한다. 아래의 그림 1은 두 개의 트랜지스터를 포함하고 있는 CMOS 반도체의 단면을 보인 것이다[7].

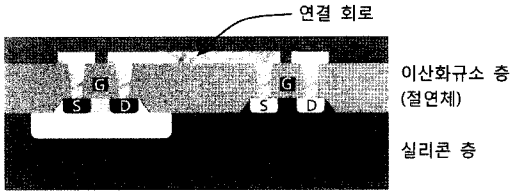


그림 1 CMOS 반도체의 단면: p형과 n형 트랜지스터

CMOS 제작 공정은 산화(oxidation), 증착(deposition), 노광(lithography), 식각(etching), 주입(implantation) 등의 다양한 과정을 반복적으로 거치며 진행된다[8]. 이러한 과정 중의 일부를 그림 2를 통하여 설명한다[7]. 그림 2는 그림 1에서 보인 CMOS 상의 트랜지스터 영역을 제조하는 최초 과정을 보인 것이다.

그림 2(a)는 CMOS 제조 공정을 시작할 실리콘 웨이퍼의 세로 단면을 보인 것이다. 그림 2(b)는 높은 온도에서 웨이퍼에 순수한 산소를 주입하여 산소와 실리콘이 반응하여 이산화규소(silicon dioxide, SiO_2)의 얇은 막이 생성된 것이다. 그림 2(c)는 이산화규소 층 위에

포토 레지스트(photo resist)의 얇은 막을 증착한 것을 보인 것이다. 그림 2(d)는 포토 마스크(photo mask)를 통하여 웨이퍼에 자외선을 비추는 노광 과정을 보인 것이다. 포토 마스크는 반도체 기술자가 웨이퍼 상에 트랜지스터 및 회로 선이 놓일 곳을 설계한 도면이다. 포토 레지스트는 자외선을 쪼이면 화학적 특성이 변하고 현상(development)하면 제거된다. 그림 2(e)는 식각 과정을 통하여 이산화규소 층의 일부가 떨어져 나간 것을 보인 것이다. 그림 2(f)는 포토 레지스트를 제거한 후의 결과를 보인 것이다. 이산화규소가 떨어져 나간 부분에 트랜지스터가 심어진다.

반도체 공정 중에 사용되는 식각 방식은 매우 다양한 방식이 있으나, 여기에서는 그림 2(e)에서 사용될 수 있는 플라즈마 식각(plasma etching) 방식에 대하여 설명한다. 그림 3은 플라즈마 식각의 과정을 개략적으로 보인 것이다[8].

그림 3에 보인 식각 공정을 정확하게 수행하기 위해서는 다양한 변수(variable) 값들을 적절하게 설정, 유지하여야 한다. 예를 들어, XeF_2 가스의 압력, Ar 플라즈마를 생성하기 위한 RF(radio frequency) 주파수, 전극 간의 전압 차 등의 다양한 변수를 관리한다. 본 논문에서 제안하는 이상탐지 알고리즘은 이러한 변수 값들을 입력으로 받아 반도체공정 중에 이상이 발생하였는지

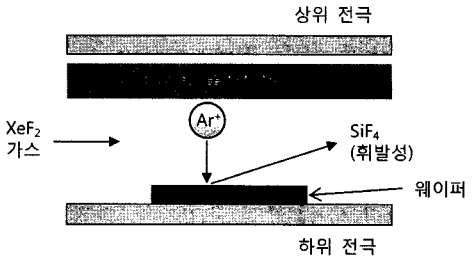
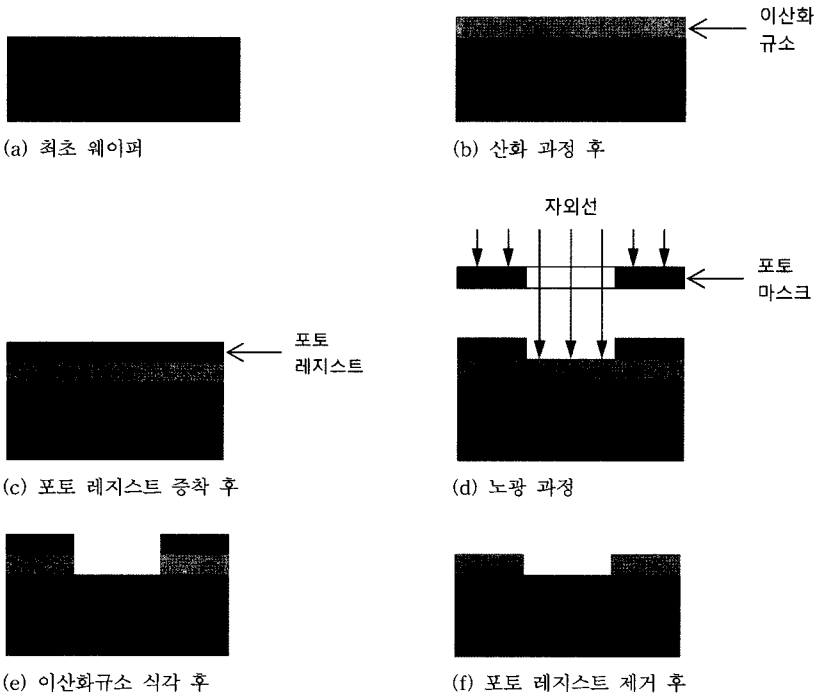


그림 3 플라즈마 식각의 개략적인 과정

탐지하고, 클러스터링 알고리즘은 이상탐지 결과를 이용하여 유사한 이상 패턴을 보이는(유사한 이상 원인을 가질 것으로 예측되는) 공정들을 분석한다.

반도체 공정에서의 FDC와 관련된 기존의 연구는 다음과 같다. 참고문헌[9]는 FDC에 대한 초기 연구로서 반응이온 식각장치(reactive ion etcher, RIE)에 실시간 피드백 제어장치를 설치함으로써 식각 성능을 크게 향상시킴을 설명하였다. 참고문헌[10]에서는 이온 주입장치(ion implanter)를 위한 분류 기반 이상탐지 알고리즘을 제안하였다. 여기에서 사용된 분류 알고리즘은 결정 트리(decision tree)[11]의 일종인 분류 및 회귀 트리(classification and regression tree, CART)를 변형한 하이브리드 분류 트리(hybrid classification tree, HCT)



(a) 최초 웨이퍼 (b) 산화 과정 후

(c) 포토 레지스트 증착 후 (d) 노광 과정

(e) 이산화규소 식각 후 (f) 포토 레지스트 제거 후

그림 2 CMOS 상의 트랜지스터 영역을 제조하는 최초 과정

이다. 참고문헌[12]에서는 Support Vector Machine (SVM) 기법을 이용하여 전력 전송 시스템(power transmission system)을 위한 이상탐지 알고리즘을 제안하였다. 또한, 웨이브렛(wavelet) 기반의 전처리를 통하여 분류 성능을 높일 수 있음을 설명하였다. 참고문헌[13]에서는 반도체 제조시설 내의 다양한 장비들로부터의 신호들 간에 필요한 시간 동기화(time synchronization)에 대하여 논하였다. 참고문헌[14]에서는 기존의 반도체 제조장비에 공정제어 시스템(process control system, PCS)을 통합하기 위한 표준을 다루고 있다. 이 표준의 내용으로 장비들 간의 통신 인터페이스와 이를 통한 이상 탐지, 이상 분류, SPC 등이 포함되어 있다.

이러한 기존의 FDC에 대한 대부분의 연구는 결정 트리, SVM, 신경망[1] 등의 통계적 또는 기계학습(machine learning) 기법들을 이용하여 수행되었다는 점이다. 많은 통계적 또는 기계학습 기법들은 그 복잡도가 높아서 매우 긴 실행 시간을 필요로 하거나 대용량의 데이터를 처리하기가 매우 어렵다. 통계적 기법은 기본적으로 대상 데이터가 특정 통계적 모델을 따름을 가정하여 문제를 해결하려 하지만, 실제 세계에서 특정 통계적 모델을 정확히 따르는 데이터는 많지 않다. 또한, 통계적 또는 기계학습 기법을 이용한 알고리즘들은 이상탐지 정확도가 100%인 경우는 존재하지 않으며, 때로 매우 낮은 경우도 존재한다. 본 연구에서는 이러한 약점들을 극복하고자 데이터베이스 분야에서의 데이터 마이닝(data mining) 기법을 이용한다.

3. 심볼릭 표현법

반도체 공정 중에 발생하는 이상을 미연에 발견하기 위하여 공정 중에 일정 시간 간격으로 다양한 측정 값들을 수집한다[1]. 이러한 측정 데이터는 기본적으로 실수의 연속으로 구성되므로 일종의 시계열 데이터이다. 반도체 공정을 비롯한 많은 응용에서 시계열 데이터의 용량이 매우 크므로, 디스크 저장 및 처리 성능을 높이기 위하여 압축된 표현법(representation)으로 변환한다[5]. 이러한 시계열 표현법으로 Discrete Fourier Transform(DFT), Discrete Wavelet Transform(DWT), Piecewise Aggregate Approximation(PAA), Adaptive Piecewse Constant Approximation(APCA), Singular Value Decomposition(SVD) 등 다양한 방법이 제안되어 왔다[15-18].

최근에는 시계열을 구성하는 연속된 실수 값을 한정된 개수의 이산 심볼로 표현하는 방법이 제안되었다. 이러한 시계열 표현법을 심볼릭 표현법이라고 하며, 기존의 실수로 구성된 시계열에 비하여 몇 가지 장점이 존재한다[5,4,6,19]. 즉, 심볼릭 표현법으로 변환된 데이터는 변환 전의 데이터에 비하여 용량이 매우 적으며, 해싱(hashing), 마르코프 모델(Markov model), 접미어 트리(suffix tree) 등 기존에 이산 심볼들(또는 알파벳)에 대하여 적용되었던 데이터 구조 및 알고리즘을 그대로 활용할 수 있다. 최근의 대표적인 심볼릭 표현법으로 Symbolic Aggregation approXimation(SAX)를 들 수 있다[6,19]. SAX는 일반적인 심볼릭 표현법의 장점 이외에 아래의 식 (1)과 같은 하한보장(lower bounding) 특성을 만족한다는 장점을 갖는다:

$$D(X,Y) \geq MINDIST(\hat{X},\hat{Y}) \quad (1)$$

여기에서, X, Y는 임의의 시계열, $\hat{X}$, $\hat{Y}$은 SAX 변환된 시퀀스, $D()$는 두 시계열 간의 거리(유사성), $MINDIST()$는 SAX 변환된 시퀀스 간의 거리를 나타낸다. 거리 함수 $D()$는 두 시계열 간의 유클리드 거리(Euclidean distance)로 정의된다. 하한보장 특성이 중요한 이유는 $MINDIST()$ 거리를 이용한 검색 결과가 부정 오류를 발생하지 않는다는 점이다.

최근에 SAX 표현법을 시계열 데이터 마이닝에 적용한 많은 연구가 발표되고 있다. 참고문헌[4,20]에서는 SAX 표현법을 이용한 불일치 탐지(discord detection) 알고리즘을 제안하였다. 참고문헌[21,22]에서는 SAX 변환된 시계열 데이터 내에서 빈번히 나타나는 패턴인 모티프(motif)를 검색하기 위한 알고리즘을 제안하였다. 참고문헌[23]에서는 SAX 표현법에 기반하여 파라미터를 최소화한 데이터 마이닝 알고리즘들을 제시하였다. 참고문헌[24]에서는 SAX 표현법을 확장하여 대용량 시계열 데이터베이스에 대한 인덱스를 효율적으로 생성하기 위한 iSAX 변환을 제안하였다.

본 논문에서는 참고문헌[4]의 불일치 탐지 알고리즘을 수정하여 반도체 공정 중의 이상을 탐지하고 클러스터링하는 알고리즘을 제안한다. 본 절에서는 알고리즘을 설명하기에 앞서 시계열 데이터를 SAX 변환하는 과정에 대하여 간략하게 설명한다. 길이 n의 시계열 X는 n개의 실수 x_1, …, x_n으로 구성되며, $X = (x_1, \cdots, x_n)$과 같이 표기한다. 주어진 파라미터 w에 대하여 시계열 X를 w개의 동일한 길이의 서브시퀀스(subsequence)로 분할하고, 각 서브시퀀스에 대하여 구성된 값들의 평균을 구한다. 시퀀스 $\overline{X} = (\overline{x}_1, \ldots, \overline{x}_w)$를 w개의 평균 값들로 구성된 시퀀스라고 한다면, 각 $\overline{x_i}$ $(1 \leq i \leq w)$는 다음의 식 (2)에 따라 구해진다:

$$\overline{x}_i = \frac{w}{n} \sum_{j=\frac{n}{w}(i-1)+1}^{\frac{n}{w}i} x_j \quad (2)$$

여기까지의 변환은 PAA 변환과 동일하다. 다음에, 시퀀스 $\overline{X}$를 정규화(normalization) 변환한다. 즉, 다음의

식 (3)에 따라 정규화 변환 시퀀스 $\overline{X}' = (\overline{x}'_1, ..., \overline{x}'_w)$을 구한다:

$$\overline{x}'_i = \frac{\overline{x}_i - \overline{\mu}}{\overline{\sigma}} \quad (3)$$

여기에서, $\overline{\mu}$는 시퀀스 $\overline{X}$를 구성하는 값들의 평균, $\overline{\sigma}$는 표준편차를 나타낸다.

정규화 변환된 시퀀스 $\overline{X}'$의 구성 값들의 평균은 0, 표준편차는 1이 되며, $\overline{x}'_i$ 값들의 분포는 0을 중심으로 정규 분포(normal distribution)를 따른다. 주어진 파라미터 a에 대하여 $\overline{x}'_i$ 값들이 w/a 개씩 균등하게 분포하도록 a 개의 영역 $R_j = [\beta_{j-1}, \beta_j)$ $(1 \le j \le a)$로 분할한다. 이때, 영역을 분할하기 위한 구분점(breakpoint) β_j는 다음의 식 (4)를 만족하도록 정해진다:

$$\int_{\beta_{j-1}}^{\beta_j} \varphi(u)du = \frac{1}{a} \quad (4)$$

여기에서, $\varphi(u)$는 정규분포 확률밀도함수(probability density function)이며, $\beta_0 = -\infty$, $\beta_a = \infty$로 정의한다. 아래의 표 1은 몇 개의 a 값에 대하여 β_j 값들을 보인 것이다.

표 1 파라미터 a 값에 따른 구분점 β_j 값들

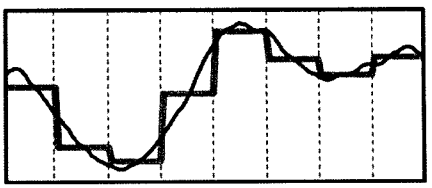
	a = 3	a = 4	a = 5	a = 6
β_1	−0.43	−0.67	−0.84	−0.97
β_2	0.43	0	−0.25	−0.43
β_3	n/a	0.67	0.25	0
β_4	n/a	n/a	0.84	0.43
β_5	n/a	n/a	n/a	0.97

최종적으로, SAX 변환된 시퀀스 $\hat{X} = (\hat{x}_1, ..., \hat{x}_w)$의 각 구성 값 $\hat{x}_i$은 다음의 식 (5)에 따라 a 개의 심볼들 중의 하나인 α_j로 정해진다:

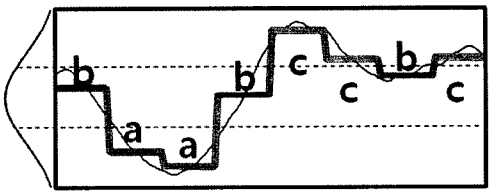
$$\hat{x}_i = \alpha_j \quad \text{iff} \quad \overline{x}'_i \in R_j \ \text{ i.e. } \beta_{j-1} \le \overline{x}'_i < \beta_j \quad (5)$$

SAX 변환된 시퀀스 $\hat{X}$을 길이 w의 워드(word)라고도 부른다.

그림 4는 길이 128인 시계열 X를 파라미터 w = 8, a = 3에 따라 SAX 변환하는 예를 보인 것이다. 그림 4(a)에서 시계열 X를 먼저 w 개의 서브시퀀스로 분할하고 각 서브시퀀스를 구성하는 값들의 평균을 구한다. 각 서브시퀀스의 길이는 n/w = 16이다. 평균값들로 구성된 시퀀스 $\overline{X}$를 정규화 변환하여 시퀀스 $\overline{X}'$을 얻는다. 그림 4(b)에서 a = 3 개의 영역으로 분할하고, 시퀀스 $\overline{X}'$의 구성 값들이 포함되는 영역에 따라 {α_1 = a, α_2 = b, α_3 = c} 중 하나의 심볼로 변환한다. 최종적으로, 시계열 X를 SAX 변환한 시퀀스 $\hat{X}$ = *baabccbc*가 얻어진다.



(a) 시계열 X를 w = 8 개의 서브시퀀스로 분할하여 각각의 평균 값을 계산



(b) a = 3 개의 영역으로 분할하고, 각 평균 값들을 하나의 심볼로 변환

그림 4 SAX 변환의 예

두 개의 SAX 변환된 시퀀스 $\hat{X}$, $\hat{Y}$ 간의 거리 *MINDIST*는 다음의 식 (6)과 같이 정의된다:

$$MINDIST(\hat{X}, \hat{Y}) = \sqrt{\frac{n}{w}\sum_{i=1}^{w}(dist(\hat{x}_i, \hat{y}_i))^2} \quad (6)$$

여기에서, n은 변환 전의 시계열의 길이, w는 SAX 변환 후의 시퀀스의 길이이다. SAX 변환 과정에서 SAX 변환 대상 시퀀스를 w 개의 서브시퀀스로 분할하고 각 서브시퀀스를 식 (2)에 따라 하나의 값으로 변환하며 그 값들을 모아 하나의 SAX 변환된 시퀀스를 생성하므로, w는 'SAX 변환 전의 서브시퀀스의 개수' 또는 'SAX 변환 후의 시퀀스의 길이'라고 표현하여도 같은 의미이다. 식 (6) 내의 *dist*() 함수는 다음의 식 (7)과 같이 정의된다:

$$dist(\alpha_s, \alpha_t) = \begin{cases} 0 & \text{if } t - s \le 1 \\ \beta_{t-1} - \beta_s & \text{otherwise} \end{cases} \quad (7)$$

여기에서, $s \le t$라고 가정하고 있으며, $dist(\alpha_t, \alpha_s) = dist(\alpha_s, \alpha_t)$이다. 아래의 표 2는 파라미터 a = 4에 대한 *dist*() 함수의 값을 보인 것이다.

표 2 a = 4에 대한 *dist*() 함수의 값

	α_1	α_2	α_3	α_4
α_1	0	0	0.67	1.34
α_2	0	0	0	0.67
α_3	0.67	0	0	0
α_4	1.34	0.67	0	0

4. 이상탐지 알고리즘

본 절에서는 먼저 이상탐지의 대상이 되는 반도체 공

정 데이터에 대하여 설명한다. 대상 데이터는 참고문헌 [1]에서 사용한 것과 동일한 데이터이며, 웨이퍼 식각 공정에서 얻어진 20개의 런(run) 데이터이다. 웨이퍼 공정은 수많은 단계를 거치는데, 이러한 단계들 중에서 연속되고 연관된 하나 이상의 공정 단계들을 하나의 런으로 구성한다. 대개 런과 런 사이에는 휴지 기간이 존재하며 기술자에 의하여 웨이퍼가 다른 장비로 이동되는 경우도 있다. 본 논문에서 사용하는 런 데이터는 10개의 모델 런(model run) 데이터와 10개의 실험 런(experimental run) 데이터로 구성된다. 모델 런은 모두 정상(nominal) 웨이퍼를 생산한 런이며, 실험 런 중에는 정상 웨이퍼를 생산한 런이 3개, 비정상(perturbed) 웨이퍼를 생산한 런이 7개 포함되어 있다. 비정상 런이 발생한 원인은 모두 다르며 표 3에 요약되어 있다.

표 3 모델 및 실험 런 데이터 설명

모델 런	실험 런	
런 번호	런 번호	설명
FDA_12	FDA_14	Unperturbed control run
FDA_16	FDA_15	−0.5mT change to base pressure
FDA_19	FDA_17	+0.5mT change to base pressure
FDA_21	FDA_20	−1% MFC conversion shift
FDA_24	FDA_23	+1% MFC conversion shift
FDA_28	FDA_25	Source RF cable: loss simulation
FDA_32	FDA_31	Unperturbed control run
FDA_37	FDA_34	Bias RF cable: power delivered
FDA_39	FDA_38	Unperturbed control run
FDA_44	FDA_43	Added chamber leak rate by 1.3mT/min

각 런 데이터는 59개의 시스템 변수(system variable)로 구성된다. 이중 처음 4개의 변수는 실제 식각 장비로부터 측정된 값이 저장되는 것이 아니라, 변수 값을 측정한 시간(0.1초 단위), 웨이퍼 이름, 레시피(recipe) 이름, 공정 단계를 나타낸다. 따라서, 이상탐지 알고리즘에서는 나머지 55개의 변수 데이터만을 이용한다. 각 변수는 총 11개의 공정 단계로 구성되고, 이중 맨 처음과 맨 마지막 단계는 각각 초기화 및 종료 단계이다. 모든 변수에 대하여 약 240초 동안의 측정 값을 포함하고 있고, 각 단계는 서로 다른 시간 간격을 갖는다. 그림 5는 모델 런 FDA_12의 51번 변수에 대하여 측정한 값들을 보인 것으로 수직 점선은 각 단계의 경계를 나타낸다.

본 논문에서 제안하는 이상탐지 알고리즘은 참고문헌 [4]의 불일치 탐지 알고리즘을 수정한 것이다. 본 논문에서는 참고문헌[4]의 알고리즘을 HOT SAX 알고리즘이라 부른다. HOT SAX 알고리즘에서는 불일치 탐지를 위하여 다음과 같은 두 가지 용어를 정의하고 있다.

정의 1. 비자신 매치(non-self match): 시계열 데이터 X에 포함된 길이 l의 서브시퀀스 S와 T가 각각 임의의 오프셋 p, q에 위치하고 있다면, 만약 $|p - q| \geq l$이 성립하면 서브시퀀스 S와 T를 비자신 매치라고 정의한다. □

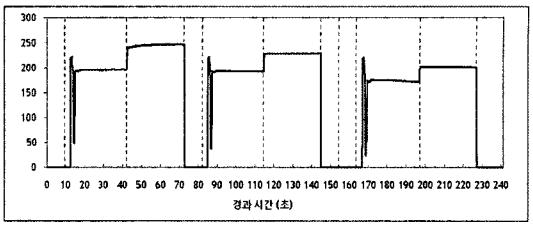


그림 5 모델 런 FDA_12의 51번 변수의 측정 값 그래프

정의 2. 불일치 서브시퀀스(discord subsequence): 시계열 데이터 X에 포함된 임의의 길이 l의 서브시퀀스 중에서 자신과 가장 거리가 가까운(유사한) 비자신 매치와의 거리가 가장 큰 서브시퀀스 S를 불일치 서브시퀀스라고 정의한다. 즉, X에 포함된 다른 모든 서브시퀀스 T에 대하여 아래의 식 (8)이 성립한다:

$$\forall T \in X, \min\{D(S, S_M)\} > \min\{D(T, T_M)\} \quad (8)$$

여기에서, S_M과 T_M은 각각 S와 T의 비자신 매치이다. □

식 (8)에서 등호를 포함하지 않는 이유는 다음과 같다. 매우 규칙적으로 변화하는 시계열 데이터에서는 등호가 성립하는 경우가 많이 발생하며, 그러한 경우들을 모두 불일치 서브시퀀스라고 볼 수 없기 때문이다.

HOT SAX 알고리즘은 주어진 임의의 시계열 X와 서브시퀀스의 길이 l에 대하여, X 내의 불일치 서브시퀀스 S를 효율적으로 검색하기 위한 알고리즘이다. HOT SAX 알고리즘은 X의 길이 n에 대하여 $O(n^2)$의 복잡도를 갖는 단순(naïve) 알고리즘에 비하여 월등한 성능을 갖는 휴리스틱(heuristic) 알고리즘이다.

본 논문에서 제안된 이상탐지 알고리즘의 목표는 정상 웨이퍼를 생산하는 모델 런 데이터를 이용하여 실험 런 중에서 비정상 웨이퍼를 생산하는 런과 그 런 내에서 문제가 발생한 변수 및 단계를 탐지해 내는 것이다. 이때, 어떤 실험 런에서 비정상 웨이퍼를 생산하는지는 사전에 알려지지 않은 것으로 가정하고 알고리즘을 설계한다. 제안된 알고리즘은 모든 공정 데이터를 각 런, 각 변수 내의 단계 단위로 분할하여 처리한다. 모델 런 데이터 중에서 동일한 변수/단계의 데이터를 모아서 실험 런 데이터의 동일한 변수/단계의 데이터와 비교한다. 만약 실험 런에서 추출한 데이터가 모델 런에서 추출한 데이터와 다르다면 이상이 발생한 것으로 간주한다.

이러한 목표를 달성하기 위하여 다음과 같이 HOT SAX 알고리즘을 수정한다. 먼저 검토하고자 하는 변수

/단계에 해당하는 데이터를 모든 모델 런으로부터 추출하여 하나의 시계열 $M_{V,S}$를 생성한다. 다음에, 이상 여부를 검토하고자 하는 하나의 실험 런으로부터 같은 변수/단계에 해당하는 데이터를 추출하여 $M_{V,S}$의 뒤에 첨부(concatenate)하여 시계열 $X_{V,S}$를 생성한다. 이 시계열 $X_{V,S}$를 입력으로 이상탐지 알고리즘을 실행한다. 그림 6은 51번 변수와 단계 2에 대하여 생성된 시계열 $X_{51,2}$를 보인 것이다.

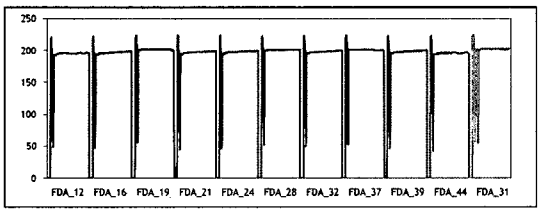


그림 6 51번 변수와 단계 2에 대하여 생성된 시계열 $X_{51,2}$

본 논문에서 제안하는 이상탐지 알고리즘과 HOT SAX 알고리즘의 차이점은 다음과 같다. 첫째, HOT SAX 알고리즘에서는 불일치 서브시퀀스의 길이 l을 입력으로 받지만, 제안된 알고리즘에서는 검토하고자 하는 변수/단계에 해당하는 시계열 데이터의 길이로 정해진다. 이 길이는 단계에 따라 달라지나, 변수가 다르더라도 단계가 동일하면 데이터의 길이는 동일하다. 둘째, HOT SAX 알고리즘에서는 반드시 하나 이상의 불일치 서브시퀀스를 검색해 내지만, 제안된 알고리즘에서는 시계열 $X_{V,S}$의 뒤쪽에 첨부된 실험 런 데이터의 불일치 여부만을 판정한다. 즉, 아래의 식 (9)가 만족되는지 검토한다:

$$\min\{D(E,M_i)\} > \max\{D(M_i,M_j)\} \quad (9)$$

여기에서, 서브시퀀스 E는 시계열 $X_{V,S}$ 내의 실험 런 데이터이며(그림 6에서 맨 마지막 붉게 나타낸 데이터: FDA_31), M_i, M_j ($i \neq j$)는 모델 런 데이터이다(그림 6에서 파랗게 나타낸 데이터: FDA_12, … , FDA_44).

식 (9)에서 서브시퀀스 E와 임의의 모델 런 데이터 M_i 간의 거리 중 최소 거리가 임의의 모델 런 데이터 쌍 (M_i, M_j) 간의 거리 중 최대 거리보다 크다는 의미는 서브시퀀스 E와 모델 런 데이터 간의 유사성이 임의의 모델 런 데이터 쌍 간의 유사성보다 작다는(서로 다르다는) 것이다. 즉, 서브시퀀스 E가 모델 런 데이터와 다르므로 이상이 발생했음을 나타내는 데이터로 판정할 수 있다. 식 (9)에서 부등호(>) 왼쪽의 항목을 D_{exp}라 하고 오른쪽의 항목을 D_{mod}라고 한다면 식 (9)는 다음의 식 (10)과 같이 쓸 수 있다:

$$D_{exp}/D_{mod} > 1.0 \quad (10)$$

본 논문에서는 D_{exp}/D_{mod} 값을 **불일치 비율**(discord ratio)이라 부르며 R_D로 표기한다. 검토 대상인 변수/단계에 대하여 불일치 비율이 1.0을 초과하면 제안된 알고리즘은 해당 변수/단계 및 런에서 이상이 발생하였다고 판정한다.

특정 변수/단계에 있어서 그 측정 값들이 거의 변화하지 않거나 거의 0인 경우가 존재하며, 이러한 경우는 표준편차가 0에 가깝고 특별하게 처리해야 한다. 제안된 알고리즘에서는 시계열 $X_{V,S}$를 생성하고 SAX 변환하는 중간에 식 (3)에 따라 표준편차를 구했을 때, 만약 그 값이 너무 작으면 해당 변수/단계에 대한 측정 데이터가 모델 런과 실험 런 내에서 거의 다르지 않다고 간주하여 이상이 없다고 판정한다.

제안된 알고리즘은 HOT SAX 알고리즘과 같이, 주어진 시계열 데이터를 SAX 변환하여 처리하고, SAX 변환된 두 서브시퀀스 간의 거리도 식 (6)에 주어진 *MINDIST*() 함수를 이용한다. 제6절에서 실험을 통하여 제안된 이상탐지 알고리즘이 반도체 공정에도 적용될 수 있음을 보인다.

5. 변수 선택 및 클러스터링 알고리즘

본 절에서는 이상탐지를 효율적으로 수행하기 위하여 하나의 런에 포함된 여러 변수들 중에서 오류가 발생하지 않는 한도 내에서 최소한의 변수를 선택하기 위한 기법과 유사한 오류 패턴을 갖는 런들의 그룹을 생성하기 위한 클러스터링 알고리즘에 대하여 설명한다.

앞의 4절에서 이상탐지 대상 데이터는 각 런에 55개의 변수에 대한 측정 값 데이터를 포함한다고 설명하였다. 이러한 모든 변수에 대한 데이터를 처리하기에는 대용량에 따른 성능 상의 문제가 있을 뿐만 아니라, 서로 다른 변수임에도 매우 유사한 형태의 측정 값을 갖는 경우도 있고, 일부 변수는 거의 모든 측정 값이 0인 경우도 있다. 따라서, 효율적인 이상탐지를 수행하기 위하여 필요한 변수만을 선택하는 기법이 필수적이다.

본 논문에서는 분류 기법 중의 하나인 교차 확인(cross validation) 기법[11]과 유사한 변수 선택 기법을 제안한다. 교차 확인 기법은 전체 데이터를 k 개의 그룹으로 분할하고, 이중 (k - 1)개의 그룹을 이용하여 분류 모델(model)을 정립하고 나머지 하나의 그룹을 이용하여 정립된 모델을 테스트한다. 분류 모델을 정립한다는 것은 주어진 분류 알고리즘을 위한 분류 데이터의 수집을 의미하며, 알고리즘 자체는 동일한 것을 사용한다. 이러한 과정을 k 번 반복하며 매 반복마다 서로 다른 그룹이 테스트 그룹이 되도록 한다.

본 논문에서 제안하는 변수 선택 기법은 먼저 표 3에서 보인 런들을 포함하는 10개의 그룹 D_i ($1 \le i \le 10$)를

생성한다. 각 그룹 D_i에는 10개의 모델 런 전체와 서로 다른 하나씩의 실험 런이 포함되며, 그룹 D_i에 포함되지 않는 런들의 그룹을 D_i*로 표기한다. 예를 들어, D_1에는 모델 런 FDA_12, …, FDA_44 전체와 실험 런 FDA_14가 하나 포함되며, D_1*에는 FDA_15, …, FDA_43이 포함된다. 다음에, 각 D_i에 대하여 모든 변수들을 이용하여 4 절에서 제안된 이상탐지 알고리즘을 수행하고 식 (10)이 만족되는(이상이 발생한) 변수들을 찾아낸다. 이때 찾아낸 변수들만을 이용하여 D_i*에 포함된 런들에 대하여 이상탐지를 수행한다.

이상탐지 알고리즘을 통하여 이상 런이 발견되면 이상이 발생한 원인에 따라 이상 런들을 클러스터링하여야 한다. 이상 런의 클러스터링이 중요한 이유는 특정 이상 런에 대하여 이상이 발생한 원인을 알지 못하더라도 그 런이 포함된 클러스터 내의 다른 이상 런과 유사한 원인일 것이라는 것을 예상할 수 있다는 점이다.

본 논문에서 제안하는 클러스터링 알고리즘은 이상 런 내에서 이상이 발생한 단계들을 이용하여 클러스터링을 수행한다. 즉, 변수와는 무관하게 동일한 단계들에서 이상이 발생된 런들을 하나의 클러스터로 구성하는 것이다. 이를 위하여 하나의 런에 대한 이상탐지 결과를 하나의 비트맵 B로 표현한다. 비트맵 $B = b_1b_2b_3b_4b_5b_6b_7b_8b_9b_{10}b_{11}$은 11개의 비트 $b_i(1 \le i \le 11)$로 구성되어 각 비트는 하나의 단계에 해당되고, 만약 해당 단계에서 이상이 발생했으면 $b_i = 1$, 그렇지 않으면 $b_i = 0$으로 설정된다. 예를 들어, 실험 런 FDA_15 내의 단계 2, 3, 4, 7에서 이상이 발생하였다면 그에 대한 비트맵 B_{15} = 01110010000이다.

제안된 클러스터링 알고리즘은 다음과 같이 실행된다. 먼저 각 실험 런 FDA_i에 대하여 자신만을 포함하는 클러스터 C_i를 생성한다. 다음에, 모든 실험 런 쌍 (FDA_i, FDA_j) ($i \neq j$)에 대하여 두 런의 이상탐지 비트맵 B_i, B_j를 비교하여, 다음의 식 (11)이 만족되면, 각 실험 런을 포함하는 두 클러스터를 하나로 통합한다:

$$Onebit(B_i \oplus B_j) \le \varepsilon \tag{11}$$

여기에서, *Onebit*() 함수는 비트맵 내의 1 비트의 개수를 반환하고, 연산자 ⊕는 XOR 연산자를 의미하고, ε은 미리 주어진 파라미터이다.

식 (11)에서 *Onebit*(…) 함수의 결과는 두 개의 런 FDA_i, FDA_j 내에서 서로 다르게 정상이거나 이상이 발생한 단계의 개수를 나타낸다. 인자 ε은 이러한 상이한 결과를 보이는 단계의 개수에 대한 한계(threshold)를 나타낸다. 하나의 런이 11 개의 단계로 구성되므로, 인자 ε은 0과 11 사이의 정수 값을 갖는다. 인자 ε은 두 런이 하나의 클러스터 내에 포함되기 위한 유사성 정의의 강도를 조절한다. 즉, 인자 ε의 값이 작아질수록 강한 유사성 정의를 나타내고, 커질수록 약한 유사성 정의를 나타낸다. 만약 인자 ε의 값이 0이면 모든 단계에서 동일한 결과를 보이는 런만이 동일한 클러스터 내에 포함되고, 인자 ε의 값이 11이면 모든 런이 하나의 클러스터 내에 포함되게 된다. 식 (11)은 ε 오차 이내에서 동일한 단계에서 이상이 발생한 런들은 유사하거나 관련 있는 원인에 의하여 이상이 발생했을 것이라는 예상에 기반한다.

6. 실험 및 평가

본 절에서는 앞에서 설명한 반도체 공정 데이터에 대한 이상탐지 및 클러스터링 알고리즘을 구현, 실험한 결과에 대하여 설명한다. 이상탐지 알고리즘에 대해서는 다음과 같은 세 가지 실험을 수행하였다. (1) Principle Component Analysis (PCA) 기법 [25]을 이용하여 변수를 선택하여 실험하였고, (2) 변수 데이터 내의 잡음(noise)을 제거하기 위한 실험을 수행하였고, (3) 5 절에서 설명한 변수 선택 기법에 따라 선택된 변수들을 이용하여 실험하였다.

본 실험에서 사용된 하드웨어는 인텔 Core2Quad CPU Q9550 2.83GHz, 4GB DDR3 메인 메모리, 640GB HDD를 장착한 PC이며, 소프트웨어는 마이크로소프트 Windows XP SP3 상에서 Visual Studio 6.0을 이용하였다.

본 논문에서는 제안된 알고리즘의 성능(실행 속도)보다는 정확성에 집중한다. 반도체 공정에서의 이상탐지에 대한 기존의 연구에서 알고리즘의 실행 시간이 턱없이 길지 않는 한 그 정확성을 훨씬 강조하고 있기 때문이다. 그 이유는 반도체 공정 중간중간에 기술자가 간여하여 많은 시간이 소요되는 반면, 이상탐지의 잘못으로 인한 공정 상의 비용 증가는 매우 크기 때문이다. 실행 시간 면에서도 변수 선택 과정을 제외하고 모든 런/단계에 대하여 제안된 알고리즘을 실행한 시간은 채 30초도 걸리지 않았다. 이 정도의 실행 시간은 이상탐지를 위하여 충분한 시간이며, 고성능의 하드웨어를 사용함으로써 더욱 개선이 가능하다.

제안된 이상탐지 알고리즘에서는 HOT SAX 알고리즘과 같이 주어진 시계열 데이터를 SAX 변환하여 처리한다. 이때 두 개의 파라미터 w와 a가 정해져야 한다. 식 (1)에서 두 시계열 데이터에 대한 *MINDIST* 거리는 항상 D 거리보다 작거나 같다고 하였다. 이때, 두 거리 간의 비율 *MINDIST*/D (≤1)는 파라미터 w와 a에 따라 변하고, 당연히 그 비율이 1에 가까울수록 좋을 것이다[6]. 파라미터 w와 a를 너무 작게 설정하면 정확도가 떨어지고, 너무 크게 잡으면 *MINDIST*/D 비율이 1에 가까워지지만 SAX 변환에 따른 성능 향상의 장점

이 줄어든다는 단점이 있다. 본 실험에서는 w = 20, a = 40으로 설정하였고, 실제로 파라미터 값이 너무 작지 않으면 대체로 일정한 실험 결과를 얻을 수 있었다.

첫번째 실험에서는 참고문헌[1]에서와 같이 PCA 기법[25]을 통하여 11개의 변수만을 선택하여 이상탐지 알고리즘을 수행하였다. 표 4는 선택된 변수와 그에 대한 설명을 정리한 것이다.

표 5는 이상탐지 알고리즘을 실행하여 각 실험 런과 단계에 대하여 얻어진 불일치 비율 R_D 값들을 보인 것이다. 특정 런과 단계에 대하여, 모든 변수에 대하여 얻어진 불일치 비율 값들 중에서 최대값을 표에 나타낸 것이다. 식 (10)에서 보인 바와 같이, R_D 값이 1보다 큰 경우에는 이상이 발생했음을 의미한다. 표 5의 결과에서 FDA_14, FDA_20, FDA_23, FDA_38 런이 정상 런이고, FDA_15, FDA_17, FDA_25, FDA_31, FDA_34, FDA_43 런이 이상 런으로 판정되었다. 하지만, 표 3에서 보인 실제 결과와 비교해 볼 때, FDA_20, FDA_23 런에 대하여 긍정 오류가 발생하였고, FDA_31 런에 대하여 부정 오류가 발생하였다.

두번째 실험은 변수 데이터 내의 잡음을 제거하기 위한 실험이다. 첫번째 실험 결과에서 FDA_31 런에 대하여 부정 오류가 발생한 원인은 그림 7에서 보인 바와 같이 정상 런의 측정 값 내에 잡음이 발생하였기 때문이다. 특정 측정 값에 대하여 그것이 잡음인지, 아니면 반도체 공정에 있어서 의미 있는, 정상/비정상적인 웨이퍼의 생산에 영향을 줄 수 있는 값인지 파악하는 것은 매우 어려운 문제이다. 실제로, 이러한 잡음을 가려내기

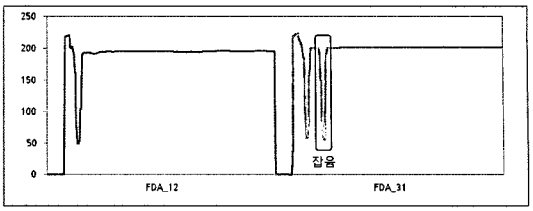


그림 7 FDA_12와 FDA_31 런의 51번 변수, 단계 2의 측정 값 비교

위해서는 정상 런과 이상 런의 측정 값들을 비교해 보는 것 외에 특별한 방법은 없다. 즉, 다른 런과 상이한 측정 값이 얻어진 경우, 그 런에서 생성된 웨이퍼의 이상 여부를 검토하여, 정상적인 웨이퍼가 생산되면 그 측정 값은 의미 없는 잡음이며, 비정상적인 웨이퍼가 생산되면 그 측정 값은 이상을 유발하는 값으로 다루어야 한다.

FDA_31 런의 단계 2와 4에 이러한 잡음 데이터가 포함되어 있어서 두 단계의 데이터에서 일괄적으로 최초 20%를 제거하고 다시 이상탐지 알고리즘을 실행하였다. 표 6은 두번째 실험 결과 얻어진 불일치 비율 R_D 값들을 보인 것이다. 두번째 실험에서는 FDA_31 런에 대하여 정상으로 판정하였다.

세번째 실험에서는 5절에서 설명한 변수 선택 기법에 따라 선택된 변수들을 이용하여 이상탐지 알고리즘을 수행하였다. 두번째 실험 결과에서 FDA_20과 FDA_23 런에 대하여 긍정 오류가 발생한 원인은 PCA를 통하여 전체 55개 중에서 11개만의 변수를 선택하여 처리한 데에

표 4 PCA 기법을 통하여 선택한 변수들

변수	설명	변수	설명
5	Pressure	21	RF probe current
7	Throttle valve percentage open	11	RF probe phase
14	Source RF match: source reading	51	RF probe peak to peak voltage
18	Source RF match: bias series	52	RF probe DC bias
19	Source RF match: bias shunt	57	E-chunk voltage
20	RF probe voltage		

표 5 첫번째 실험 결과: 이상탐지 알고리즘의 실험 결과 얻어진 불일치 비율

	1	2	3	4	5	6	7	8	9	10	11	판정
FDA_14	0.176	0.868	0.761	0.099	0.430	0.462	0.000	0.000	0.163	0.488	0.000	정상
FDA_15	0.175	7.114	5.300	2.829	0.450	0.463	1.429	0.000	0.722	0.490	0.097	이상
FDA_17	0.201	7.310	5.334	0.217	0.468	0.408	3.346	0.000	1.362	0.797	0.449	이상
FDA_20	0.051	0.377	0.818	0.041	0.572	0.336	0.000	0.245	0.223	0.687	0.143	정상
FDA_23	0.841	0.353	0.748	0.000	0.415	0.607	0.000	0.000	0.326	0.554	0.068	정상
FDA_25	0.028	0.794	2.544	2.536	4.667	44.406	3.101	3.101	4.163	50.580	0.037	이상
FDA_31	0.000	1.706	0.869	5.648	0.502	0.361	0.000	0.000	0.554	0.718	0.543	이상
FDA_34	0.036	1.190	3.790	2.326	2.347	68.041	3.361	3.361	1.574	84.181	0.297	이상
FDA_38	0.000	0.976	0.665	0.041	0.405	0.368	0.000	0.000	0.593	0.610	0.092	정상
FDA_43	0.614	0.523	0.702	0.000	1.301	0.978	1.542	1.542	4.143	1.252	0.111	이상

표 6 두번째 실험 결과: 이상탐지 알고리즘의 실험 결과 얻어진 불일치 비율

	1	2	3	4	5	6	7	8	9	10	11	판정
FDA_14	0.176	0.402	0.761	0.178	0.430	0.462	0.000	0.000	0.163	0.488	0.000	정상
FDA_15	0.175	6.381	5.300	0.817	0.450	0.463	1.429	0.000	0.722	0.490	0.097	이상
FDA_17	0.201	6.447	5.334	0.387	0.468	0.408	3.346	0.000	1.362	0.797	0.449	이상
FDA_20	0.051	0.500	0.818	0.269	0.572	0.336	0.000	0.245	0.223	0.687	0.143	정상
FDA_23	0.841	0.567	0.748	0.152	0.415	0.607	0.000	0.000	0.326	0.554	0.068	정상
FDA_25	0.028	1.222	2.544	2.536	4.667	44.406	3.101	3.101	4.163	50.580	0.037	이상
FDA_31	0.000	0.697	0.869	0.135	0.502	0.361	0.000	0.000	0.554	0.718	0.543	정상
FDA_34	0.036	1.398	3.790	2.326	2.347	68.041	3.361	3.361	1.574	84.181	0.297	이상
FDA_38	0.000	0.648	0.665	0.233	0.405	0.368	0.000	0.000	0.593	0.610	0.092	정상
FDA_43	0.614	0.602	0.702	0.205	1.301	0.978	1.542	1.542	4.143	1.252	0.111	이상

있다. PCA 기법은 가장 최적의 변수를 선택할 수 있도록 하지만, 실제로 오류가 발생한 런들에 대하여 고려하지 않으므로 이상탐지 알고리즘에서 긍정 오류 또는 부정 오류가 발생할 가능성이 있으며, 대용량 데이터에 대해서 실행 시간이 매우 오래 걸린다는 단점이 있다[5,24].

본 논문에서 제안된 변수 선택 기법을 실행한 결과 얻어진 36, 37, 38번 변수를 새로이 추가하여 제안된 이상탐지 알고리즘을 실행하였다. 표 7은 추가된 변수와 그에 대한 설명을 정리한 것이다. 이들 변수들은 서로 연관된 변수들로서 이들 중 하나 또는 둘 만을 추가하여도 정확한 결과를 얻을 수 있다. 표 8은 세번째 실험 결과 얻어진 불일치 비율 R_D 값들을 보인 것이다. 실험 결과, 긍정 오류 및 부정 오류가 전혀 발생하지 않았다.

부록에서 세번째 실험 결과에 따라 각 단계 및 변수별로 불일치 비율 값들을 점으로 표시한 그림을 보인다. 연속된 실험의 결과를 통해 제안된 이상탐지 알고리즘이 정확하게 이상을 탐지할 수 있음을 알 수 있다.

마지막으로, 세번째 실험 결과를 이용하여 제5절에서 제안한 클러스터링 알고리즘을 실행하였다. 이때 파라미터 ε = 1로 설정하였고, 클러스터링 결과 얻어진 실험 런 그룹들은 다음과 같다:

- 그룹 1: { FDA_14, FDA_31, FDA_38 } (정상 런)
- 그룹 2: { FDA_15, FDA_17 }
- 그룹 3: { FDA_20, FDA_23 }
- 그룹 4: { FDA_25, FDA_34 }
- 그룹 5: { FDA_43 }

실험 런들에 대한 설명을 표 3에서 보면, 같은 그룹에 포함된 실험 런들은 유사한 이상 원인을 갖고 있음을 알 수 있다. 따라서, 제안된 클러스터링 알고리즘을 통해 유용한 결과를 얻을 수 있음을 알 수 있다.

7. 결 론

본 논문에서는 반도체 공정 중의 하나인 식각 공정에서 얻어진 측정 값들을 이용하여 이상탐지 및 클러스터링을 수행하기 위한 알고리즘을 제안하였다. 제안된 알고리즘은 참고문헌[4]의 HOT SAX 알고리즘을 수정한 것이다. 제안된 알고리즘에 대한 실험 결과, SAX 표현법에 기반한 이상탐지 알고리즘이 반도체 공정 데이터

표 7 변수 선택 기법을 통하여 추가된 변수들

변수	설명	변수	설명
36	Flow splitter: flow 1	38	Flow splitter: total flow
37	Flow splitter: flow 2		

표 8 세번째 실험 결과: 이상탐지 알고리즘의 실험 결과 얻어진 불일치 비율

	1	2	3	4	5	6	7	8	9	10	11	판정
FDA_14	0.176	0.402	0.761	0.332	0.373	0.430	0.487	0.422	0.000	0.000	0.163	정상
FDA_15	0.175	6.381	5.300	0.817	0.311	0.450	0.463	1.429	0.000	0.000	0.722	이상
FDA_17	0.201	6.447	5.334	0.387	0.322	0.468	0.455	3.346	0.000	0.000	1.362	이상
FDA_20	0.051	0.500	2.151	2.267	2.366	2.569	3.216	2.653	0.000	0.245	0.223	이상
FDA_23	0.841	0.567	2.335	1.898	2.371	2.711	3.149	2.635	0.000	0.000	0.326	이상
FDA_25	0.028	1.222	2.001	2.544	2.536	4.667	4.719	44.406	3.101	3.101	4.163	이상
FDA_31	0.000	0.697	0.869	0.336	0.513	0.502	0.400	0.274	0.000	0.000	0.554	정상
FDA_34	0.036	0.539	1.398	3.790	2.326	2.347	1.006	68.041	3.361	3.361	1.574	이상
FDA_38	0.000	0.648	0.665	0.233	0.338	0.405	0.430	0.299	0.000	0.000	0.593	정상
FDA_43	0.614	0.602	0.702	0.386	0.451	1.301	0.978	0.900	1.542	1.542	4.143	이상

에 대해서도 정확한 결과를 얻을 수 있으며, 유용한 클러스터를 생성함을 보였다.

앞으로의 연구 방향의 하나로 반도체 공정 데이터 내에 포함된 잡음 데이터를 자동으로 찾아서 제거하는 방법에 대한 연구가 필요할 것이다. 잡음 제거에 대한 연구는 특정 반도체 공정에 종속적인 경우가 많다. 6절에서도 언급한 바와 같이, 특정 데이터 값이 잡음인지 아닌지는 해당 반도체 공정에서 이상이 발생하였는지 여부를 확인하여야 한다. 특정 데이터 값이 잡음인지 자동적으로 해석하기 위한 이론은 현재까지 존재하지 않으며, 많은 연구소 및 기업에서 그 주제에 대하여 연구가 진행되고 있다.

참 고 문 헌

[1] S. J. Hong, G. S. May, J. Yamartino, and A. Skumanich, "Automated Fault Detection and Classification of Etch Systems Using Modular Neural Networks," In *Proc. of the SPIE*, vol.5378, pp.134-141, Feb. 2004.

[2] D. C. Montgomery, *Introduction to Statistical Quality Control*, 6th Edition, Wiley, May 2008.

[3] G. G. Barna, "APC in the semiconductor industry, history and near term prognosis," In *Proc. of the Advanced Semiconductor Manufacturing Conference and Workshop (ASMC)*, IEEE/SEMI, pp.364-369, Nov. 1996.

[4] E. Keogh, J. Lin, and A. Fu, "HOT SAX: Efficiently Finding the Most Unusual Time Series Subsequence," In *Proc. of the IEEE Int'l Conf. on Data Mining (ICDM)*, Houston, Texas, pp. 226-233, Nov. 2005.

[5] H. Ding, G. Trajcevski, P. Scheuermann, X. Wang, and E. Keogh, "Querying and Mining of Time Series Data: Experimental Comparison of Representations and Distance Measures," In *Proc. of the VLDB Endowment (PVLDB)*, vol.1, no.1, pp.1542-1552, Aug. 2008.

[6] J. Lin, E. Keogh, S. Lonardi, and B. Chiu, "A Symbolic Representation of Time Series, with Implications for Streaming Algorithms," In *Proc. of the ACM SIGMOD Workshop on Research Issues in Data Mining and Knowledge Discovery (DMKD)*, San Diego, California, pp.2-11, June 2003.

[7] SEMI, *Silicon Run I*, 2nd Edition, Training Video, Silicon Run Productions, 1996.

[8] H. J. Hwang, *Semiconductor Process Technology*, Saeng-Reung Publishers, July 2000. (in Korean)

[9] B. A. Rashap et al., "Control of Semiconductor Manufacturing Equipment: Real-Time Feedback Control of a Reactive Ion Etcher," *IEEE Trans. on Semiconductor Manufacturing*, vol.8, no.3, pp.286-297, Aug. 1995.

[10] S.-Y. Lin and S.-C. Horng, "A Classification-Based Fault Detection and Isolation Scheme for the Ion Implanter," *IEEE Trans. on Semiconductor Manufacturing*, vol.19, no.4, pp.411-424, Nov. 2006.

[11] J. Han and M. Kamber, *Data Mining: Concepts and Techniques*, Morgan Kaufmann, Second Edition, Nov. 2005.

[12] O. A. S. Youssef, "An optimised fault classification technique based on Support-Vector-Machines," In *Proc. Power Systems Conference and Exposition (PES)*, pp.1-8, Mar. 2009.

[13] V. Anandarajah et al., "Precise Time Synchronization in Semiconductor Manufacturing," In *Proc. of IEEE Int'l Symposium on Precision Clock Synchronization for Measurement, Control and Communication (ISPCS)*, Vienna, Austria, pp.78-84, Oct. 2007.

[14] J. R. Moyne, H. Hajj, K. Beatty, and R. Lewandowski, "SEMI E133-The Process Control System Standard: Deriving a Software Interoperability Standard for Advanced Process Control in Semiconductor Manufacturing," *IEEE Trans. on Semiconductor Manufacturing*, vol.20, no.4, pp.408-420, Nov. 2007.

[15] K. Chan and A. W. Fu, "Efficient Time Series Matching by Wavelets," In *Proc. of the IEEE Int'l Conf. on Data Engineering (ICDE)*, Sydney, Australia, pp.126-133, Mar. 1999.

[16] C. Faloutsos, M. Ranganathan, and Y. Manolopoulos, "Fast Subsequence Matching in Time-Series Databases," In *Proc. of the ACM SIGMOD Int'l Conf. on Management of Data,* Minneapolis, Minnesota, pp.419-429, May 1994.

[17] P. Geurts, "Pattern Extraction for Time Series Classification," In *Proc. of the European Conf. on Principles of Data Mining and Knowledge Discovery*, Freiburg, Germany. pp.115-127, Sep. 2001.

[18] E. Keogh, K. Chakrabarti, M. Pazzani, and S. Mehrotra, "Locally Adaptive Dimensionality Reduction for Indexing Large Time Series Databases," In *Proc. of ACM SIGMOD Conf. on Management of Data*, Santa Barbara, California, pp.151-162, May 2001.

[19] J. Lin, E. Keogh, L. Wei, and S. Lonardi, "Experiencing SAX: a Novel Symbolic Representation of Time Series," *Data Mining and Knowledge Discovery (DMKD)*, vol.15, no.1, pp.107-144, Aug. 2007.

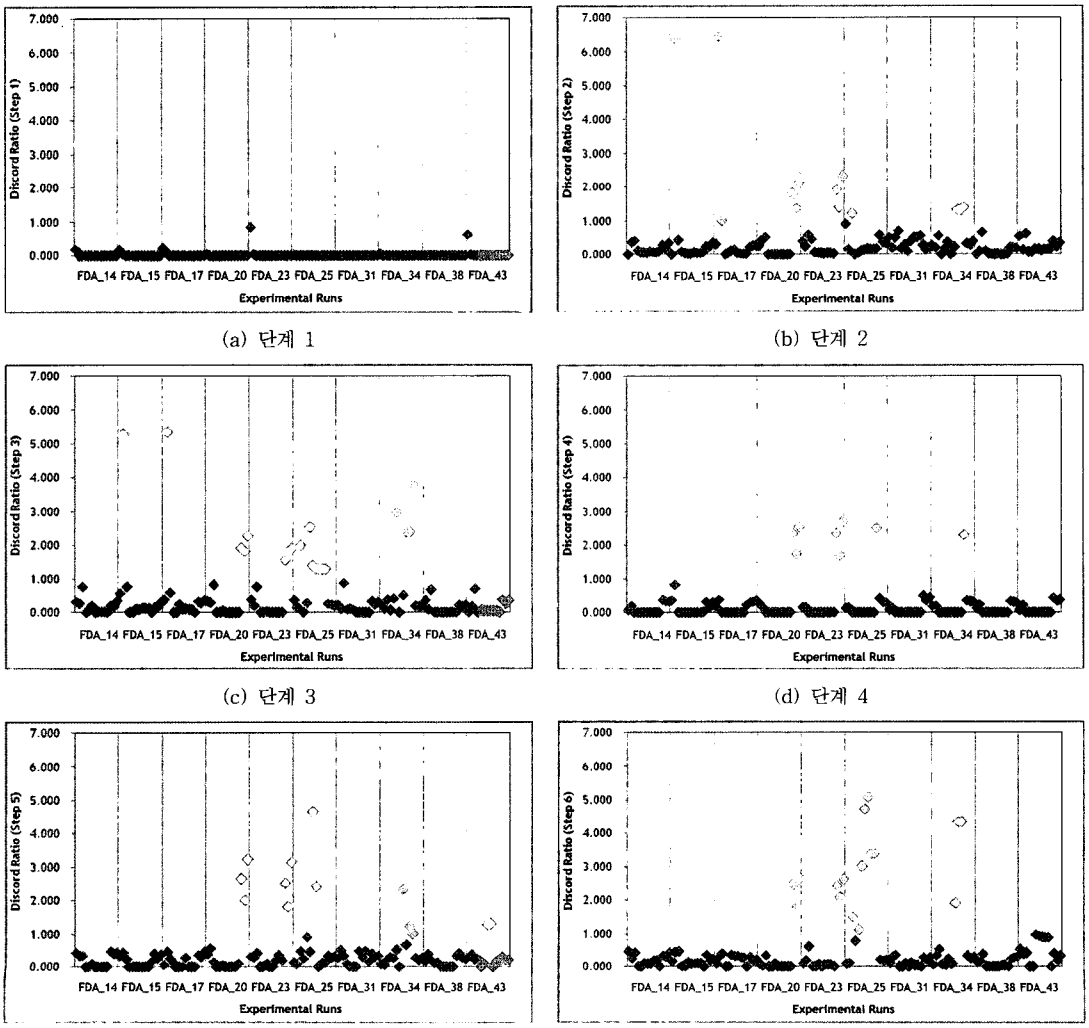
[20] L. Wei, N. Kumar, V. N. Lolla, E. Keogh, S. Lonardi, and C. A. Ratanamahatana, "Assumption-Free Anomaly Detection in Time Series," In *Proc. of the Int'l Scientific and Statistical Database Management Conf. (SSDBM)*, Santa Barbara, California, pp.237-240, June 2005.

[21] B. Chiu, E. Keogh, and S. Lonardi, "Probabilistic Discovery of Time Series Motifs," In *Proc. of the ACM SIGKDD Int'l Conf. on Knowledge Disco-*

very and Data Mining, Washington DC, USA, pp. 493-498, Aug. 2003.
[22] P. Patel, E. Keogh, J. Lin, and S. Lonardi, "Mining Motifs in Massive Time Series Databases," In *Proc. of the IEEE Int'l Conf. on Data Mining (ICDM)*, Maebashi City, Japan, pp. 370-377, Dec. 2002.
[23] E. Keogh, S. Lonardi, and C. Ratanamahatana, "Towards Parameter-Free Data Mining," In *Proc. of the ACM SIGKDD Int'l Conf. on Knowledge Discovery and Data Mining*, Seattle, Washington, pp.206-215, Aug. 2004.
[24] J. Shieh and E. Keogh, "iSAX: Indexing and Mining Terabyte Sized Time Series," In *Proc. of the ACM SIGKDD Int'l Conf. on Knowledge Discovery and Data Mining*, Las Vegas, Nevada, pp.623-632, Aug. 2008.
[25] W. L. Martinez and A. R. Martinez, *Exploratory Data Analysis with MATLAB*, Chapman & Hall, Nov. 2004.

부 록

그림 8은 각 단계 및 변수별로 불일치 비율 값들을 점으로 표시한 것이다. 각 그림에서 하나의 실험 런에 대하여 모든 변수의 불일치 비율 값들을 점으로 표시하였고, 이중 1 이상인 붉은 점으로 표시된 값들은 이상이 발생하였음을 나타낸다.

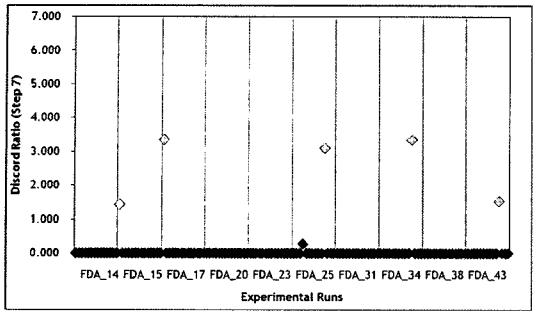


(a) 단계 1 (b) 단계 2

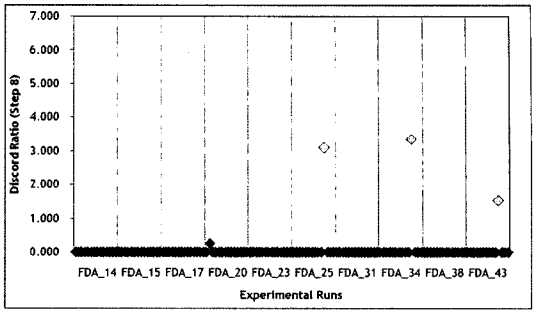
(c) 단계 3 (d) 단계 4

(e) 단계 5 (f) 단계 6

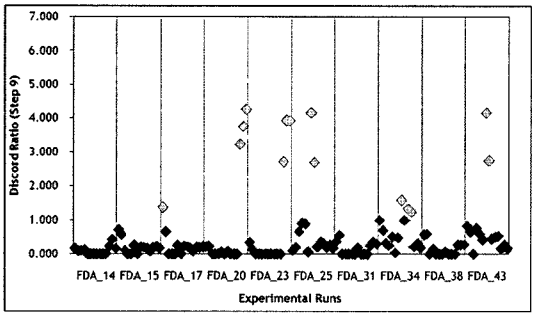
그림 8 단계 및 변수별 불일치 비율 값들 (계속)



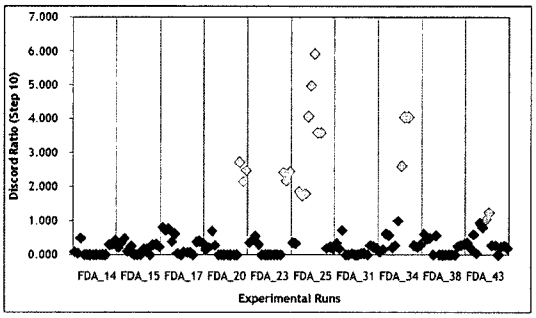
(g) 단계 7



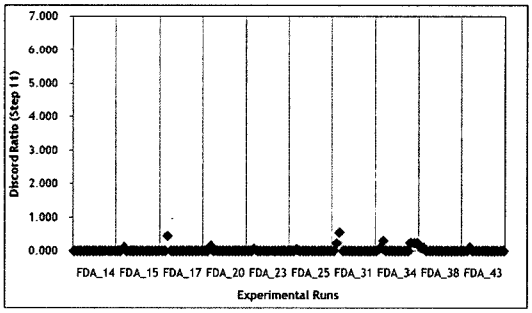
(h) 단계 8



(i) 단계 9



(j) 단계 10



(k) 단계 11

그림 8 단계 및 변수별 불일치 비율 값들

노 웅 기

1991년 2월 한국과학기술원(KAIST) 전산학과 학사. 1993년 2월 한국과학기술원(KAIST) 전산학과 석사(멀티미디어 전공). 2001년 2월 한국과학기술원(KAIST) 전산학과 박사 (데이터 마이닝 전공). 2001년 2월~2003년 9월 ㈜티맥스소프트 책임연구원(미들웨어 개발). 2003년 10월~2005년 3월 ㈜티맥스데이타 수석연구원(DBMS 엔진 개발). 2005년 4월~2006년 5월 한국과학기술원 전산학과 초빙교수. 2006년 6월~2007년 7월 Visiting Scholar, University of Minnesota, USA. 2007년 8월~2008년 2월 NHN㈜ 수석연구원(대용량 로그 데이터 분석). 2008년 3월~현재 성결대학교 멀티미디어학부 전임강사. 관심분야는 대용량 데이터 마이닝, 데이터 웨어하우징, 정보 검색, 멀티미디어 데이터베이스, 멀티미디어 정보 검색, 데이터베이스 시스템 엔진

홍 상 진

1999년 2월 명지대학교 전기전자공학부 제어계측전공 공학사. 1999년 8월~2001년 5월 Georgia Institute of Technology, M.S. 2001년 5월~2001년 8월 IBM, East FishKill, Process Engineer. 2001년 8월~2003년 12월 Georgia Institute of Technology, Ph.D. 2004년 1월~2004년 8월 일본 동북대학교 외국인특별연구원(JSPS Fellow). 2004년 8월~현재 명지대학교 전자공학과 부교수. 관심분야는 반도체공정개발 및 선행공정제어(APC), 센서기반 공정진단