

Cluster-Based Sequence Analysis of Complex Manufacturing Process

Kittisak Kerdprasop and Nittaya Kerdprasop, *Member, IAENG*

Abstract—Wafer fabrication in the semiconductor industry is probably one of the most complex manufacturing processes. Maintaining high yields through the statistical process control as a sole monitoring method for quality control is obviously inefficient in such highly complicated operations. We thus present in this paper a sequence analysis method, which is one of the advanced data mining techniques, to identify and extract unique patterns from manufacturing data that can reveal and differentiate low performance processes from the normal ones. We also provide the program coding, implemented with the R language, for easy experimental repetition.

Index Terms—Sequence analysis, sequence patterns, data mining, computer-aided manufacturing, clustering, R language

I. INTRODUCTION

MANY modern manufacturing process engineers are now facing with the problem of timely yield analysis from large scale and multi-dimensional data that are automatically generated daily from hundreds of operational units in the production line. Semiconductor manufacturing is one of the most highly complex production processes in which they are composed of hundreds of steps. The major tasks in most semiconductor industries are: production of silicon wafers from pure silicon material, fabrication of integrated circuits onto the raw silicon wafers, assembly by putting the integrated circuit inside a package to form a ready-to-use product, and testing of the finished products [9]. A constant advancement in the semiconductor industry is due mainly to persistent improvement of the wafer fabrication process.

The fabrication process consists of a series of steps to cover special material layers over the wafer surface. Wafers re-enter the same processing machines as each layer is successively covered. Some defects in this complicated process can make the final products fail the test. Early fault detection during this critical manufacturing process can obviously improve product quality and reliability.

Manuscript received December 7, 2012; revised January 10, 2013. This work was supported in part by grant from Suranaree University of Technology via the funding of Data Engineering Research Unit, in which the authors are principal researchers.

Kittisak Kerdprasop is an associated professor with the School of Computer Engineering, Suranaree University of Technology, 111 University Avenue, Nakhon Ratchasima 30000, Thailand (e-mail: kittisakThailand@gmail.com).

Nittaya Kerdprasop is an associate professor and head of the Data Engineering Research Unit, School of Computer Engineering, Institute of Engineering, Suranaree University of Technology, 111 University Avenue, Nakhon Ratchasima 30000, Thailand (phone: +66-(0)44-224432; fax: +66-(0)44-224602; e-mail: nittaya@sut.ac.th).

Recent trend in intelligent manufacturing is to apply the variety of data mining techniques to automatically identify patterns and causal relationships leading to poor yield. In this paper, we expand the frontier of data mining application to the manufacturing process area by proposing an advanced sequence data mining technique. Our proposed technique can be viewed as a compliment of the classical statistical process control in that it can help engineers detecting process variations in a semi-automatic manner.

Sequence is an ordered set of elements in which each element can be numerical, categorical, or a mixture of attributes. The order of elements could be determined by their occurring time or positions. If the order is by time and the elements of a sequence are real values, it is a time series. When the sequence elements are discrete, it is a categorical sequence [13]. Sequence mining is a recently active field of research in knowledge discovery and data mining. The applications of the available techniques are mostly in the areas of bioinformatics and financial analysis. In this paper, we demonstrate the potential application of sequence data mining to discover the operational sequences of tools causing low yields in the complex manufacturing process.

II. RELATED WORK

In recent years, many manufacturing tools are equipped with sensors to facilitate real-time monitoring of the production process. These tool-state and production-state sensor data provide an opportunity for efficient control and optimization. Unfortunately, such measurement data are so overwhelming that timely detection of any fault during the production process is difficult. Therefore, automatic and advanced process control method is required.

Ison and colleagues [7] proposed a decision tree classification model to detect fault of plasma etch equipment. The model was built from the five sensor signal data. Goodlin *et al* [4] proposed to build a specific control chart for detecting specific type of faults. They collected tool-state data directly from the etcher. These data consist of 19 variables. The work of Spitzlsperger and colleagues [14] was also based on the statistical method. They adopted the multivariate control chart method to maintain changes in the mean and standard deviation coefficients by remodeling technique.

Later interest in fault detection has been shifted toward the non-parametric approaches. He and Wang [5] proposed to use the k-nearest neighbor rule for fault detection. Verdier and Ferreira [16] also applied the k-nearest neighbor method, but they proposed to use the adaptive Mahalanobis distance instead of the Euclidean distance.

Tafazzoli and Saif [15] proposed a combined support vector machine methodology for process fault diagnosis. Ge and Song [3] applied support vector data to the principal component analysis method to detect process abnormalities.

Most work on fault detection methods has studied the process control problem with a few features of tool-state and process-state measurement data. McCann and his team [10] proposed a rather different setting in which the measurement data from the wafer fabrication process contain as much as 590 features. They applied feature selection technique to select only 40 features for further analysis.

In this work, we apply a data mining technique that can handle 150 features of sequential data, rather than independent and discrete data as proposed in all the previous work. Sequence data mining of manufacturing process appeared in the literature just a few years ago [11], [12]. Our work presented in this paper is different from others in that we apply sequence analysis as an exploratory tool, instead of the classification tool. Moreover, we adopt the open source paradigm for the purpose of reproducible experimentation.

III. A CLUSTER-BASED SEQUENCE ANALYSIS METHOD

A. Description of Manufacturing Process Data

In our sequence analysis, we use the data named SETFI (SEmiconductor Tool level Fault Isolation) simulated by Advanced Analytics, Intel [1]. This dataset closely emulates the actual highly complex nature of most semiconductor manufacturing processes. The dataset contains 4000 records of wafer lots, each lot went through hundreds of operations along the wafer fabrication process. During the process each wafer goes through sequence of operations in batch, which is called lot in this dataset. The sequences of operations might be different from lot to lot, but they involve only twenty kinds of equipments or tools, numbering from 1 to 20. Tool distribution in the wafer fabrication process is graphically shown in Fig.1. At each operational unit, however, only a single tool is in operation.

At the end of the fabrication process, a number of inspection steps are carried out to measure the product performance. Wafer lots that fail the inspection tests need re-processing. These low quality products require special attention. Low performance metric is often caused by a small subset of tools. Identifying such problematic tools at an early stage can obviously improve yield performance of the semiconductor manufacturing. Some data instances of the SETFI dataset are shown in Table 1. The first data instance contains information of a wafer lot number 3699 that starts the process with a tool number 2 and ends with a tool number 3. Its performance metric is 2841.763.

The original dataset has 300 operational units, each unit has a timestamp of the operation. In this study, we ignore the timestamps as our main objective is categorical sequence analysis, not a time series. We also remove the first column (Lot#) because it plays no role to the discovered sequence patterns. According to the data source document, missing values in this dataset are around 25%.

TABLE 1
DATA EXAMPLES OF SOME WAFER LOTS

Lot#	Op_1	...	Op_300	Performance
3699	2	...	3	2841.763
1427	9	...		2779.744
	2	...	3	2721.452
...
1753	7	...	5	2732.957

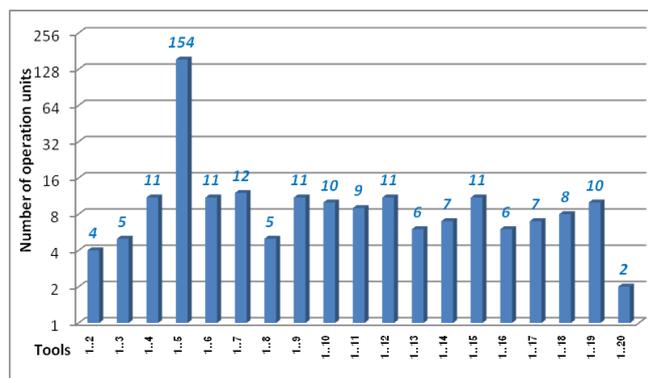


Fig.1 Distribution of tools applied in the wafer operational units

B. A Cluster-Based Sequence Mining Technique

From the manufacturing process dataset that contains information of 300 operational units of 4000 wafer lots, we firstly perform data-preprocessing. The data preparation for our analysis method starts with the extraction of 301 features (or variables) containing the tools used in the 300 operational units plus the performance metric, which is the last column in the SETFI dataset. Then imputing missing information with the NA constant value, followed by the data exploration. We investigate our data characteristic by clustering the data with the pamk() function that is available in the fpc package of R language [6], [17]. This function is an extension of the partitioning around medoid algorithm [8] that also suggests appropriate number of clusters considering from the optimum average silhouette width. The pamk() function returns two clusters as follows:

- Cluster 1: contains 2028 records,
 minimum performance = 2781.227,
 maximum performance = 3293.183.
- Cluster 2: contains 1972 records,
 minimum performance = 2177.438,
 maximum performance = 2783.146.

From all 4000 wafer lots, the maximum performance is 3293.183, whereas the minimum one is 2177.438. The mean performance value is 2787.924. Therefore, it can be seen from this preliminary result that the wafer fabrication process dataset can be divided based on the performance metrics into two disjoint groups: a group of wafers that are approximately above average (cluster 1), and another group of those that are below average (cluster 2). We call cluster 1 a group of high lots, and cluster 2 a group of low lots. We then design the analysis framework (as shown in Fig.2) to differentiate these two groups, and also to compare against the average group of wafer lots.

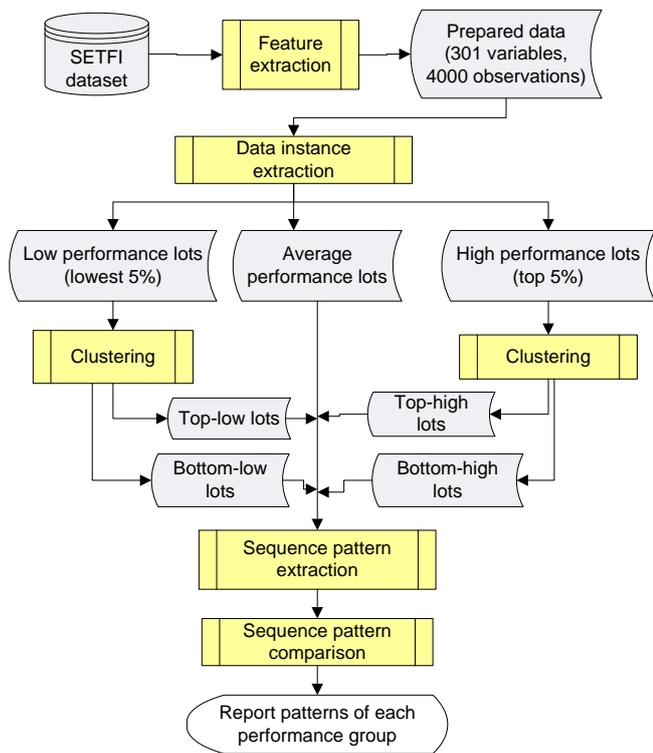


Fig.2 A Framework of cluster-based manufacturing process sequence analysis

After the data instance extraction step (in Fig.2), all 4000 wafer lots are separated into three groups: low, median, and high performance groups. The number of wafer lots in each group are around 200 to 201 data instances. Performance characteristics (maximum, minimum, mean, and standard deviation) of these three groups are summarized in Table 2. The tasks of data preparation, clustering, and sequence pattern extraction are performed through the R commands with the coding provided in Fig.3. To run the program, users call the main function: `mainTC()`. This is the only function in our program.

The SETFI dataset is in a file 'com.csv'. The first command in the program is to read the data and store in the variable 'dat'. The first column, which is the lot number, is then removed. Then the dataset has been sorted in descending order according to the performance value. The ordered data of 4000 wafer lots are called 'dat3all'. This dataset is then divided into four subsets, i.e., 'low', 'mid', 'high', and 'all', and stored in the variable named 'my.dat'. The performance metric in the last column is then attached to the dataset, and call this new dataset 'cdat'. It is then clustered by the `pamk()` function, and the result is in the variable 're'. The program calls `seqdef`, `seqcreate`, and `seqfsub` functions from the library TraMineR [2].

Default parameter of the function `mainTC()` is the 'low' data. To extract sequence patterns from other subsets, the parameter has to be specified. For example, `mainTC(high)` is to extract patterns from a group of wafer lots with high performance. The parameters 'from' and 'to' are for identifying data columns to be analyzed. Parameter 'per' is a percentage to split data into high, low, and median groups. The last parameter is minimum support, 'min', in which 0.7 has been set as default.

TABLE 2
PERFORMANCE CHARACTERISTICS OF WAFER LOTS

Wafer Lots	Performance			
	Maximum	Minimum	Mean	S.D.
Low (200 lots)	2574.012	2177.438	2503.816	66.95
Median (201 lots)	2790.671	2778.334	2784.345	3.70
High (201 lots)	3293.183	2992.259	3062.469	63.95
All (4000 lots)	3293.183	2177.438	2787.924	125.84

```
library(TraMineR)
library(fpc)
mainTC <- function(from=1, to=300, lot='low',
                    per=0.05, min=0.7)
{ dat <- read.csv('com.csv'); dat <- dat[-1]
  dat3 <- dat
  dat3all <- dat3[with(dat3, order(res)),]
  my.dat <- switch(lot,
                  low = dat3all[1:(per*nrow(dat3))],
                  mid = dat3all[(0.5*nrow(dat3)-
                                (per*nrow(dat3))/2):(0.5*nrow(dat3)+
                                (per*nrow(dat3))/2-1),],
                  high = dat3all[((1-per)*nrow(dat3)):4000,],
                  all = dat3all)
  #-----
  cat('\n*****\nPerformance, max=',
      max(my.dat[601]),'min=',min(my.dat[601]),'\n')
  # use 300 cols -- and use performance col 601.
  cdat <- my.dat[,c(1:300,601)] ; re <- pamk(cdat)
  my.ind <- re$pamobject$clustering
  # print each cluster
  cat(lot,'lot +++++ has',re$nc,'clusters\n
      cluster #0=all data in this lot\n')
  for (i in 0:re$nc)
  { if (i==0) {data.used <- cdat}
    else
      {data.used <- cdat[my.ind==i,] }
    mvad.seq <- seqdef(data.used, var=from:to,
                      missing=NA)
    # Event sequence analysis
    mvad.seqe <- seqcreate(mvad.seq)
    fsubseq <- seqfsub(mvad.seqe, pMinSupport=min,
                      maxK=3)
    cat('-----\nCluster#',i,'of',lot,
        'lot--Max Performance=',max(data.used[301]),',
        min=', min(data.used[301]),'of',
        nrow(data.used) ,'records\n')
    print(fsubseq[1:50])
    # plot the 15 most frequent sequences
    plot(fsubseq[1:15],
         main = paste('Cluster#',i,'has ',nrow(data.used),
                     'records at',lot,'lot ,Columns :Col',
                     from,'-','to','--Max Performance=',
                     max(data.used[301]),',min=',
                     min(data.used[301])) )
  }
}
```

Fig. 3 R programming for data clustering and sequence pattern extraction

IV. SEQUENCE ANALYSIS RESULTS

To analyze the wafer fabrication lot patterns, we have to run the sequence extraction program four times, varying a lot parameter as 'low', 'mid', 'high', and 'all' in each execution. Users may call these executions in one time and save the output in a file 'out.txt' with the following commands:

```
sink('out.txt')
for(t in c('low', 'mid', 'high', 'all')) { mainTC(lot=t) }
sink()
```

The results of all executions that were written in a file 'out.txt' can be illustrated in Fig.4. Each discovered subsequence is attached with the support and count values. The interpretation of these subsequence extraction results is straightforward. For instance, the first subsequence in a low performance lot is that tool number 2 had been applied prior to the tool number 5 (displayed as 2>5) with a frequency count of 200 from the total 200 records in this group. That means this subsequence support value is a proportion of 200/200, or equal to 1.000 as shown in Fig.4. The possible maximum support value is 1, whereas the minimum is 0. The higher this support value, the more frequent a sequence had occurred.

```
*****
Performance, max= 2574.012 min= 2177.438
low lot +++++ has 2 clusters
cluster #0=all data in this lot
-----
Cluster# 0 of low lot--Max Performance= 2574.012
,min= 2177.438 of 200 records
Subsequence Support Count
1 (2>5) 1.000 200
2 (3>1) 1.000 200
3 (3>4) 1.000 200
4 (3>4)-(4>5) 1.000 200
...
49 (2>5)-(4>5) 0.990 198
50 (2>5)-(5>1) 0.990 198
Computed on 200 event sequences
Constraint Value
countMethod COBJ
-----
Cluster# 1 of low lot--Max Performance= 2500.938
,min= 2177.438 of 74 records
Subsequence Support Count
1 (1>4) 1 74
2 (1>4)-(4>3) 1 74
...
49 (3>2)-(5>3)-(3>1) 1 74
50 (3>2)-(5>3)-(3>2) 1 74
Computed on 74 event sequences
Constraint Value
countMethod COBJ
-----
Cluster# 2 of low lot--Max Performance= 2574.012
,min= 2500.235 of 126 records
Subsequence Support Count
1 (1>2) 1.0000000 126
2 (1>2)-(1>2) 1.0000000 126
...
-----
Cluster# 2 of all lot--Max Performance= 3293.183
,min= 2781.227 of 2028 records
Subsequence Support Count
1 (3>2) 0.9995069 2027
2 (2>5) 0.9990138 2026
...
50 (3>4)-(1>2) 0.9876726 2003
Computed on 2028 event sequences
Constraint Value
countMethod COBJ
```

Fig.4 Sequence extraction results from all groups of wafer lots

The focus of our sequence analysis is the ability to differentiate frequent manufacturing process patterns of low performance lots from the high performance lots. From the clustering results, we concentrate our analyses on the bottom low lots (containing 74 records) that show the lowest performances comparative to the top high lots (with 46 records) that show the highest performances. The sequences of these two groups are shown in Fig.5. Long chaining subsequences such as (1>3)-(3>2)-(2>1) can be interpreted as the sequence of tool number 1 followed by tool number 3 used in the two operational units is normally preceding the other two tool sequences, that is, 3→2 and 2→1, respectively. In our experimentation, we set the number of sequence items to be at most three (due to the memory space limitation), and to display only the top-50 sequences. All sequences in Fig.5 have support value 1.

(Low) Subsequence	(High) Subsequence
1 (1>4)	1 (1>3)
2 (1>4)-(4>3)	2 (1>3)-(2>1)
3 (1>5)	3 (1>3)-(2>3)
4 (1>5)-(2>4)	4 (1>3)-(3>2)
5 (2>3)	5 (1>3)-(3>2)-(2>1)
6 (2>3)-(3>1)	6 (1>3)-(3>2)-(2>3)
7 (2>3)-(3>2)	7 (1>3)-(3>2)-(3>2)
8 (2>3)-(4>3)	8 (1>3)-(3>2)-(5>3)
9 (2>4)	9 (1>3)-(3>4)
10 (2>4)-(2>4)	10 (1>3)-(3>4)-(2>1)
11 (2>4)-(2>5)	11 (1>3)-(3>4)-(3>2)
12 (2>4)-(2>5)-(2>4)	12 (1>3)-(4>1)
13 (2>4)-(2>5)-(5>2)	13 (1>3)-(5>2)
14 (2>4)-(3>1)	14 (1>3)-(5>2)-(2>1)
15 (2>4)-(3>2)	15 (1>3)-(5>3)
16 (2>4)-(3>4)	16 (1>4)
17 (2>4)-(5>2)	17 (1>4)-(1>3)
18 (2>4)-(5>3)	18 (1>4)-(1>3)-(3>2)
19 (2>5)	19 (1>4)-(2>1)
20 (2>5)-(1>5)	20 (1>4)-(2>1)-(3>2)
21 (2>5)-(2>3)	21 (1>4)-(2>3)
22 (2>5)-(2>4)	22 (1>4)-(2>3)-(3>2)
23 (2>5)-(2>4)-(3>2)	23 (1>4)-(2>4)
24 (2>5)-(3>2)	24 (1>4)-(2>5)
25 (2>5)-(5>2)	25 (1>4)-(2>5)-(2>3)
26 (3>1)	26 (1>4)-(2>5)-(3>2)
27 (3>1)-(2>3)	27 (1>4)-(2>5)-(4>5)
28 (3>1)-(4>2)	28 (1>4)-(3>2)
29 (3>1)-(5>2)	29 (1>4)-(3>2)-(4>2)
30 (3>2)	30 (1>4)-(3>5)
31 (3>2)-(1>4)	31 (1>4)-(4>1)
32 (3>2)-(2>3)	32 (1>4)-(4>1)-(2>3)
33 (3>2)-(2>3)-(3>1)	33 (1>4)-(4>2)
34 (3>2)-(2>3)-(3>2)	34 (1>4)-(4>2)-(2>1)
35 (3>2)-(2>4)	35 (1>4)-(4>2)-(2>4)
36 (3>2)-(2>4)-(3>2)	36 (1>4)-(4>2)-(4>5)
37 (3>2)-(3>1)	37 (1>4)-(4>5)
38 (3>2)-(3>1)-(5>2)	38 (1>4)-(4>5)-(3>2)
39 (3>2)-(3>2)	39 (1>4)-(4>5)-(4>2)
40 (3>2)-(3>2)-(5>3)	40 (1>4)-(4>5)-(5>3)
41 (3>2)-(3>4)	41 (1>4)-(5>3)-(2>1)
42 (3>2)-(4>3)	42 (1>4)-(5>3)-(3>2)
43 (3>2)-(5>1)	43 (1>4)-(5>3)-(4>2)
44 (3>2)-(5>1)-(3>1)	44 (2>1)-(2>1)
45 (3>2)-(5>1)-(5>3)	45 (2>1)-(1>3)
46 (3>2)-(5>2)	46 (2>1)-(1>3)-(3>2)
47 (3>2)-(5>3)	47 (2>1)-(1>3)-(5>3)
48 (3>2)-(5>3)-(2>4)	48 (2>1)-(1>4)
49 (3>2)-(5>3)-(3>1)	49 (2>1)-(1>4)-(2>1)
50 (3>2)-(5>3)-(3>2)	50 (2>1)-(1>4)-(2>3)

Fig. 5 Frequently occurred tool sequences of low performance lots (left) compared to those in high performance lots (right)

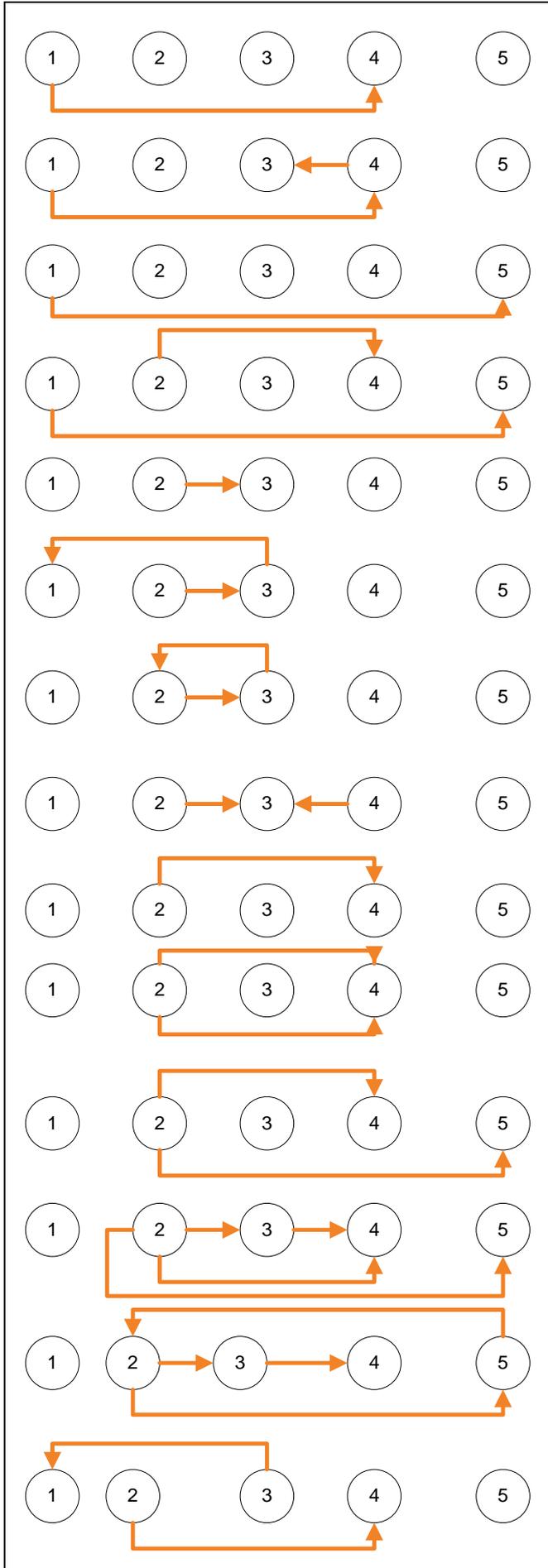


Fig. 6 Top-14 sequences of low performance lots

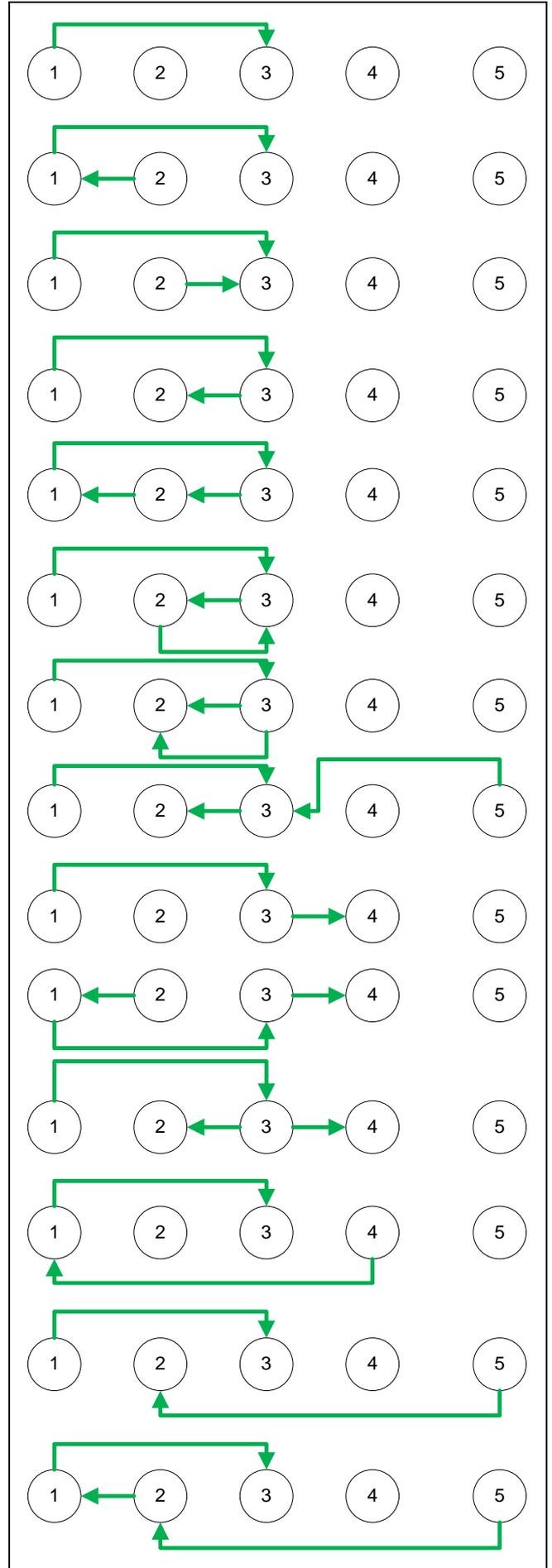


Fig. 7 Top-14 sequences of high performance lots

To compare the highly occurred tool sequences of low performance lots against the high performance lots, we graphically draw the diagrams (Figs.6 and 7) of top-14 tool sequences. From the diagrams, we can notice that the top performance lots involve sequences of tools 1,2, and 3, whereas the low performance sequences involve the tools 1 and 5. We then decompose the chaining sequences of the top-50 sequences in the three groups, that are low, median, and high performances, down to a single sequence to find a unique sequence in the low performance as well as the high performance group. The outcome is shown in Fig. 8. We then draw the experimental conclusion that a unique pattern in the low performance group is a sequence of tool 5→1, and a unique pattern in the high performance group is 1→3.

Low	Median	High
		1→3
1→4		1→4
1→5	1→5	
	2→1	2→1
2→3	2→3	2→3
2→4		2→4
2→5	2→5	2→5
3→1	3→1	
3→2	3→2	3→2
3→4	3→4	3→4
	3→5	3→5
	4→1	4→1
4→2	4→2	4→2
4→3	4→5	4→5
5→1		
5→2	5→2	5→2
5→3	5→3	5→3

Fig. 8 Unique pattern comparison among low, median, and high performance groups

V. CONCLUSIONS

For most highly complex manufacturing processes such as semiconductor industries, hundreds of metrology data are available for process engineers to analyze for the purpose of maintaining efficient operations and getting optimum yield of high quality products. For such a large volume of measurement data, automatic data analysis technique is essential. We thus investigate the application of advanced mining technique, namely sequence data mining, to help analyzing problematic sequences in the wafer fabrication process of semiconductor industries.

We designed an analysis framework to group operational process data into three categories: processes with low, high, and moderate performance metrics. Process data in each category were then analyzed with the sequence mining program written in R language. We found from the experimental results that the frequently occurred sub-sequences of each category show unique patterns. Sequence analysis technique presented in this paper is semi-automatic in the sense that unique pattern inspection has to be done by human. We thus plan to further our research towards the

design and implementation of an automatic tool to timely detect process trends leading to low performance products.

REFERENCES

- [1] AA&YA, Intel (2008, November 24). SETFI: Manufacturing data: Semiconductor tool fault isolation. *Causality Workbench Repository*. [Online]. Available: <http://www.causality.inf.ethz.ch/repository.php>
- [2] A. Gabadinho, G. Ritschard, M. Studer, and N. Muller, "Extracting and rendering representative sequences," in A. Fred, J.L.G. Dietz, K. Liu, and J. Filipe (eds.), *Knowledge Discovery, Knowledge Engineering and Knowledge Management. Series: Communications in Computer and Information Science (CCIS)*, vol. 128, pp.94-106, Springer, 2011.
- [3] Z. Ge and H. Song, "Semiconductor manufacturing process monitoring based on adaptive substatistical PCA," *IEEE Transactions on Semiconductor Manufacturing*, vol.23, no.1, pp.99-108, 2010.
- [4] B.E. Goodlin, D.S. Boning, H.H. Sawin, H.H., and B.M. Wise, "Simultaneous fault detection and classification for semiconductor manufacturing tools," *Journal of The Electrochemical Society*, vol.150, no.12, pp.G778-G784, 2003.
- [5] Q.P. He and J. Wang, "Fault detection using the k-nearest neighbor rule for semiconductor manufacturing processes," *IEEE Transactions on Semiconductor Manufacturing*, vol.20, no.4, pp.345-354, 2007.
- [6] C. Hennig. (2012, August 30), Package 'fpc': Flexible procedures for clustering, version 2.1-4. [Online]. Available: <http://www.homepages.ucl.ac.uk/~ucakche/>
- [7] A.M. Ison, W. Li, and C.J. Spanos, "Fault diagnosis of plasma etch equipment," in *Proceedings of IEEE International Symposium on Semiconductor Manufacturing*, San Francisco, pp.B-49-B-52, 1997.
- [8] L. Kaufman and P.J. Rousseeuw, *Finding Groups in Data: An Introduction to Cluster Analysis*, Wiley, 1990.
- [9] G.S. May and C.J. Spanos, *Fundamentals of Semiconductor Manufacturing and Process Control*. John Wiley & Sons, 2006.
- [10] M. McCann, Y. Li, L. Maguire, and A. Johnston, "Causality challenge: benchmarking relevant signal components for effective monitoring and process control," in *Proceedings of JMLR Workshop*, Canada, pp.277-288, 2008.
- [11] L. Rokach, R. Romano, and O. Maimon, "Mining manufacturing databases to discover the effect of operation sequence on the product quality," *Journal of Intelligent manufacturing*, vol. 19, pp.313-325, 2008.
- [12] N. Ruschin-Rimini, O. Maimon, and R. Romano, "Visual analysis of quality-related manufacturing data using fractal geometry," *Journal of Intelligent manufacturing*, vol. 23, pp.481-495, 2012.
- [13] S. Sarawagi, "Sequence data mining," in S. Bandyopadhyay, U. Maulik, L.B. Holder, and D.J. Cook (eds.), *Advanced Methods for Knowledge Discovery from Complex Data*, pp.153-187, Springer, 2005.
- [14] G. Spitzlsperger, C. Schmidt, G. Ernst, H. Strasser, and M. Speil, "Fault detection for a via etch process using adaptive multivariate methods," *IEEE Transactions on Semiconductor Manufacturing*, vol.18, no.4, pp.528-533, 2005.
- [15] E. Tafazzoli and M. Saif, "Application of combined support vector machines in process fault diagnosis," in *Proceedings of American Control Conference*, St. Louis, pp.3429-3433, 2009.
- [16] G. Verdier and A. Ferreira, "Adaptive Mahalanobis distance and k-nearest neighbor rule for fault detection in semiconductor manufacturing," *IEEE Transactions on Semiconductor Manufacturing*, vol.24, no.1, pp.59-68, 2011.
- [17] Y. Zhao. (2012, June 8). R and Data Mining: Examples and Case Studies. [Online]. Available: <http://www.rdatamining.com>