This article has been accepted for publication in IEEE Transactions on Semiconductor Manufacturing. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TSM.2022.3201654

Commonality Analysis for Detecting Failures Caused by Inspection Tools in Semiconductor Manufacturing Processes

Dae Woong An, Seung Kim, Hyun Kyu Kim, and Chang Ouk Kim

***Abstract*— Semiconductor fabrication involves hundreds of process steps through various manufacturing tools. These processing steps are composed of many manufacturing and inspection steps. Inspection is an important step in the fabrication process to determine whether a process is in or out of control. Abrupt manufacturing or inspection tool excursion can lead to a serious low yield problem. Although commonality analysis is a proven tool for detecting abrupt tool excursion, it has gained only limited success in detecting manufacturing tool excursion outside of inspection tools. Compared with manufacturing tools, only a small number of lots or wafers pass through inspection tools. Therefore, it is difficult to construct a sufficient lot history log for inspection commonality analysis in contrast to that of manufacturing tools. Furthermore, inspection may stress a wafer during its own processing, therefore, the target wafer is changed sequentially or randomly. Accordingly, a lot history is apt to include missing traces, which hinders finding inspection tool excursion effectively. In this paper, we propose a comparative analysis framework for commonality analysis algorithms. Performance measures are suggested. To compare the performance of the algorithms effectively, we use a synthetically generated dataset in a simulation experiment. In addition, we apply the algorithms to a real problem that occurred in the fabrication process. Our proposed algorithm demonstrates superiority over the other commonality analysis algorithms in the experiments.**

***Index Terms*— Commonality Analysis; Apriori; Cumulative Factor; PLS-VIP, Random Forest, STUCCO, TAR2**

I. Introduction

THE semiconductor chip manufacturing process involves hundreds of processing steps. The entire process is composed of a few repeating unit processes: thin film, photolithography, chemical mechanical planarization, diffusion, ion implantation and etching. This nature of semiconductor manufacturing leads to long cycle times [1]. As the complexity of the process and the number of process steps increase, it is definitely a very challenging task to pinpoint which tool is the source of a problem and at which process step it occurs [2]. Commonality analysis (CA) refers to a set of techniques to identify systematic causes of yield loss. Although in-line process control and statistical process control play a large role in controlling excursion [3], a low-level intermittent problem is often not detectable in-line but is only found when yield/electrical parametric data are collected later [4]. Taking advantage of electric lot history logs, systematic tool commonality among bad lots is thus a proven technique to identify the root cause of a problem [5], [6].

Table I (A) illustrates a typical wafer history log. Wafers pass through sequentially in each step during processing. Each row in the table corresponds to a wafer, but it may be a lot instead of an individual wafer. Each column's field value represents which tool processed each wafer at the corresponding step. For example, in step 1, tool 129 processed wafer 1. The last column is a numerical value of each wafer's yield. This can be discretized using appropriate binning methods. Table I (A) can be converted to a step-tool (or attribute-value) pair representation [7], as illustrated in (B). In this representation, the columns are filled with binary digits. Each of these features represents a pair, a step and tool combination. The actual value will be binary, true (1) or false (0), indicating whether that tool was used at that step. The last column can also be discretized, e.g., 'Good' or 'Bad', or 'Normal' or 'Abnormal' under a user given threshold.

TABLE I
AN EXAMPLE OF WAFER HISTORY LOG

(A) NOMINAL TYPE

Wafer-ID	Step_1	Step_2	…	Step_500	Yield
1	Tool_129	Tool_229	…	Tool_577	70
2	Tool_229	Tool_229	…	Tool_542	92
…	…	…	…	…	…
1000	Tool_129	Tool_202	…	Tool_577	79

(B) STEP-TOOL PAIR TYPE

Wafer-ID	Step 1-Tool 129	Step 1-Tool 229	…	Step 500-Tool 542	Yield
1	1	0	…	0	70
2	0	1	…	1	92
…	…	…	…	…	…
1000	1	0	…	0	79

Semiconductor fabrication involves hundreds of manufacturing and inspection steps. Due to the consideration of operation cost and efficiency, only a few wafers per lot are

This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIT) under Grant NRF-2022R1A2B5B01001889. (*Corresponding author*: *Chang Ouk Kim*.)

D. W. An, and H. K. Kim are with Data Science, SK Hynix, Icheon 17336, Republic of Korea.

S. Kim is currently with DX Division, LG Chem., Seoul 07336, Republic of Korea. This paper was written when he was in SK Hynix.

C. O. Kim is with the Department of Industrial Engineering, Yonsei University, Seoul 03722, Republic of Korea (e-mail: kimco@yonsei.ac.kr).

inspected. Usually, the number can vary from only 2% to approximately 5%. If we are concerned about inspection tool excursion, we can try to CA with a wafer history log, which is composed of inspection step columns. However, the height of the history log table, i.e., the number of records, shrinks as much as the small inspection portion. This is the first difficulty of a CA inspection tool. Another difficulty is the missing value problem [8]. An inspection target wafer is not identical through a continuous inspection step. Continuous inspection on an identical wafer may damage it by itself. Therefore, the inspection target wafer is changed sequentially or randomly at each inspection step. If we construct a wafer history log, many cells of the table may have missing values. The last difficulty is class imbalance [9]. This causes a significant bottleneck in the performance attainable by standard learning methods, which assume a balanced distribution of the classes. In an inspection tool CA, it is also a major bottleneck in performance. The above three difficulties hinder trying CA with inspection tools. Therefore, we need a more effective and robust algorithm to cope with these difficulties [10].

After 'Good' or 'Bad' discretization of the continuous valued target column on the dataset in Table I (A), we can construct a classification model (i.e., a random forest (RF), classification and regression tree (CART), support vector machine (SVM)) and assess the variable importance of the model. A high importance variable, which has high discriminant power, distinguishes between 'Good' and 'Bad' and may indicate root causes of yield loss. This naïve approach is effective if there is no severe class imbalance. However, traditional classification methods do not work well on usual highly imbalanced data [11]. We give a detailed impact of the class imbalance ratio on various algorithms in Chapter 3.

The primitive goal of CA looks like that of association rule mining, but association rule mining algorithms (i.e., Apriori [12], FP-growth [13], ECLAT [14], DHP [15], CLOSET [16]) are not suitable for this problem [11], [17]. Association rule mining finds rules whose support is above some minimum thresholds. Although we want to find the rules about low-yield wafers, these rules on a minor class are lowly supported and easily ignored.

Contrast set mining is a good alternative that pinpoints which tool is the source of a problem. It is a form of association rule mining that seeks to identify meaningful differences between separate groups by reverse engineering the key predictors that identify each particular group. The problem of mining contrast sets was first defined in [17] as finding contrast sets as "conjunctions of attributes and values that differ meaningfully in their distributions across groups". A contrast set is a conjunction of attribute–value pairs that is equivalent to an itemset in association rule mining. The Search and Testing for Understandable Consistent Contrasts (STUCCO) algorithm proposed in the original contrast set mining paper [17] discovers a set of contrast sets along with their supports on groups. Our proposed algorithm is based on the STUCCO algorithm, which is described in more detail in Chapter 2.

Treatment learning is a form of weighted contrast-set learning that takes a single desirable group and contrasts it against the remaining undesirable groups (the level of desirability is represented by weighted classes) [18], [19], [20]. The resulting treatment suggests a set of rules that, when applied, will lead to the desired outcome. The treatment learner TAR2 is compared with our proposed algorithm in Chapter 3.

In [21], the authors use the partial least squares with variable importance in projection (PLS-VIP) and Apriori algorithms. Because the number of inline FAB steps is so large to treat using the Apriori algorithm directly, the authors use PLS-VIP as a candidate selector, which filters out nonsuspicious faulty tools based on PLS-VIP scores. In this paper, the PLS model uses a step-tool pair dataset, as illustrated in Table I (B). They suggest a single factor and cumulative factor. A single factor is a measurement calculated for each tool and represents a suspicious statistical metric of low yield cause. The cumulative factor is calculated for two or more tool combinations and represents how much yield worsens when these tools are correlated. We borrow the cumulative factor concept and devise and use similar measures in our proposed algorithm.

The rest of the paper is organized as follows. In the next chapter, a comprehensive literature review is performed. In Chapter 2, our proposed algorithm is presented. Then, in Chapter 3, a comparative analysis framework for commonality analysis algorithms is explained with performance measures. The dataset generation method is presented and used in a simulation experiment. A real problem case that occurred in the fabrication process is presented at the end of the chapter. Finally, we conclude this paper in Chapter 4.

II. PROPOSED APPROACH

We assume that history logs are composed of not only manufacturing tools but also inspection tools. Inspection follows a few repeating unit processes: thin film, photolithography, chemical mechanical planarization, diffusion, ion implantation and etching. We named this repeating unit a step chunk.

Table II shows a history log scheme. In this scheme, all steps

TABLE II
HISTORY LOG SCHEME

Wafer-ID	…	$\text{Step\_}(i, j, k)$	$\text{Step\_}(i, j+1, k)$	…	$\text{Step\_}(i, j+m, k)$	…	Yield	Binned_Yield
1	…	'Tool_129'	'Tool_239'	…	Null	…	70	'Bad'
2	…	'Tool_129'	'Tool_210'	…	Null	…	92	'Good'
3	…	'Tool_132'	'Tool_239'	…	'Tool_100'	…	94	'Good'
4	…	'Tool_141'	'Tool_210'	…	Null	…	72	'Bad'
…	…	…	…	…	…	…	…	…
N	…	'Tool_129'	'Tool_210'	…	'Tool_200'	…	73	'Bad'

This article has been accepted for publication in IEEE Transactions on Semiconductor Manufacturing. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TSM.2022.3201654

are represented by triple $(\mathrm{i}, \mathrm{j}, \mathrm{k})$. Each wafer has its yield as a continuous value in the yield column. It can be binned by a certain user-given threshold. N is the total number of wafers. A step chunk is composed of serial manufacturing and its final inspection step. The whole table shows repetitions of a number of step chunks. A sequential index of the step chunk is represented by index i. A sequential index of the tools in a step chunk is represented by index j. Index k (0 or 1) is a binary value indicating manufacturing or inspection tools. The columns are filled in with a string or null value. Unique strings represent tools. We assume that there can be many null values in the inspection step.

Under this log scheme, we try to find an excursion of the inspection tool. Our algorithm is based on the χ2 significance test, which is also used in the STUCCO algorithm. We focus on only a relative degraded excursion, not an absolute excursion degree. Therefore, we propose a modified cumulative factor (MCF) that represents a relative degraded excursion caused by the inspection tool. The algorithm is composed of two steps. In the first step, it generates all candidate contrast sets. Support of the contrast set is defined with respect to each 'Bad' or 'Good' yield group. The support of a contrast set X with respect to a 'Bad' (or 'Good') group is the percentage of examples in the 'Bad' (or 'Good') group for which contrast set X is true (denoted as support(X,'Bad') or support(X,'Good')). The candidate contrast set does not just include inspection; it also includes manufacturing. Each candidate contrast set X is checked to determine whether it satisfies Equation (1) and Equation (2). δ is a user-defined threshold called the minimum-deviation. Another user-defined p-value should be given to determine whether it satisfies Equation (2).

Since we want to find only the tool that has a negative effect on yield, we do not use the abs function in for Equation (1). Contrast sets for which Equation (1) is statistically supported are called large, and those for which Equation (2) is satisfied are called significant. Table III shows a contingency table to test satisfying Equation (2) for a contrast set X.

$$support(\mathrm{X}, '\mathrm{Bad}') - support(\mathrm{X}, 'Good') > \delta. \quad (1)$$

$$p(\mathrm{X}, 'Bad') \neq p(\mathrm{X}, 'Good'). \quad (2)$$

TABLE III
CONTINGENCY TABLE FOR A CONTRAST SET X

	'Bad'	'Good'	Total
X	$support(\mathrm{X}, '\mathrm{Bad}')$	$support(\mathrm{X}, '\mathrm{Good}')$	$support(\mathrm{X})$
¬X	$support(\neg\mathrm{X}, '\mathrm{Bad}')$	$support(\neg\mathrm{X}, '\mathrm{Good}')$	$support(\neg\mathrm{X})$
Total	$support('\mathrm{Bad}')$	$support('\mathrm{Good}')$	

X means that ¬X does not occur. support(¬X) is the fraction of all records that do not contain X. If a contrast set X is independent of the yield, then we would expect the proportion of the support(X,'Bad') to the support(¬X,'Bad') and the proportion of the support(X,'Good') to the support(¬X,'Good') to be roughly equal. The standard test for the independence of variables in the contingency table is the χ2 significance test. It works by computing the χ2 statistic:

$$\chi^2 = \sum_{i=1}^{r} \sum_{j=1}^{c} \frac{(O_{ij} - E_{ij})^2}{E_{ij}}, \quad (3)$$

where O_{ij} is the observed frequency count in cell ij, and E_{ij} is the expected frequency count in cell ij given independence of the row and column variables and is calculated as $E_{ij} = \sum_j O_{ij} \sum_i O_{ij} / N$.

In the second step, in contrast to [23], the MCF is calculated for only the inspection candidate set. Precisely speaking, an MCF is the degree of degraded excursion caused by a pure inspection tool. Therefore, the calculation is performed between the manufacturing contrast set and inspection contrast set in each chunk. Each contrast set in a chunk has accuracy, which is the fraction of 'Bad' wafers containing the contrast set. If an abrupt excursion occurs in the inspection tool, the accuracy of the contrast set of the preceding inspection tool will increase significantly compared to the contrast set of the preceding manufacturing tools. The MCF is defined in Equation (4).

$$MCF(X)$$

$$= \frac{Increase\ in\ accuracy\ by\ the\ manufacturing\ operation}{Accuracy\ of\ the\ inspection\ operation}$$

$$= \max_{j \in precedings} \left(\frac{accuracy(A_j \cap X) - accuracy(A_j)}{accuracy(A_j)} \right)$$

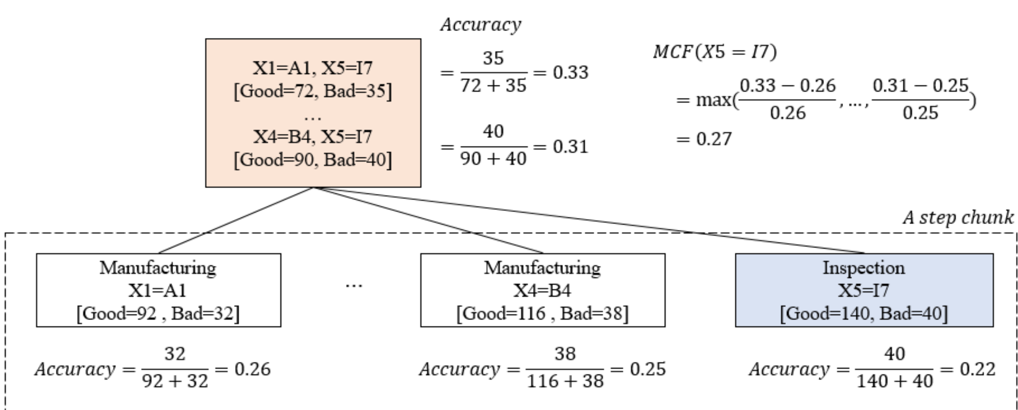


Fig. 1. An MCF calculation example.

$$= \max_{j \in precedings} \left(\frac{support(A_j \cap X, 'Bad') - support(A_j, 'Bad')}{support(A_j, 'Bad')} \right), \tag{4}$$

where A_j is a manufacturing contrast set preceding inspection contrast set X. Figure 1 represents an example of an MCF calculation. In this example, the MCF of the inspection contrast set X5 = I7 is calculated as 0.27. This means that the I7 tool at the X5 inspection step increases the accuracy by at least 27% from its preceding manufacturing steps.

In Figure 2, our proposed algorithm is summarized.

Our proposed algorithm adopts the same structure as the STUCCO algorithm with MCF. We define a contrast set to be significant or statistically significant if it passes a χ^2 test. To determine if the differences in proportions are significant, we first pick a test α level. We define a contrast set to be large or practically significant if there exist two groups such that the difference of the average values of the contrast set in these two groups is greater than some user-defined threshold δ. Then, the MCF of the contrast set is computed.

Algorithm Commonality Analysis for Inspection Equipment
Input: data *D*,
    binary threshold θ,
    minimum-deviance δ,
    significant level α
Output: O
Begin
Define 'Good', 'Bad' yield group from data D using θ
Define attribute set $A = \{A_1, A_2, \dots, A_i\}$ for all steps except yield from data D
Define equipment value set $V_i = \{V_{i1}, V_{i2}, \dots, V_{ij}\}$ for each attribute set A_i
Define contrast set C in conjunction with attribute and value set
Set of Candidates $P \leftarrow \emptyset$
for each $X \in C$ **do**
    make contingency table for X
    compute χ^2 static return *p-value*
    compute $support(X, 'Bad') - support(X, 'Good')$ return *dev*
    if *p-value* $< \alpha$, $dev > \delta$ and X is inspection **then** Compute $\mathtt{MCF}(X)$
    if $\mathtt{MCF}(X) > 0$ **then** $P \leftarrow X$
$O \leftarrow$ SortValue(MCF, *p-value*, *dev*)
End

Fig. 2. Commonality Analysis for the Inspection Tool algorithm.

III. Comparative Analysis

In this chapter, we conduct a comparative analysis of CA algorithms, including our proposed algorithm. To evaluate the performance of the algorithms, we use a synthetically generated history log.

A. Synthetically generated history log

If all manufacturing and inspection tools in the fabrication process are in normal status, there is only a small and negligible variation through all tools in a process. We call this a process in-control. However, if an uncontrollable excursion occurs, there also happens a bias in its history log. We can intentionally impose a certain bias in certain tools to some degree in a synthetically generated history log. We use a synthetically generated history log are as follows:

- First, an excursion will not happen frequently in a real FAB situation. All efforts to eliminate the occurrence of an excursion are made in real situations. To evaluate algorithm performance in various parameters, a real-world dataset will not be sufficient.
- Second, a synthetically generated history log may not fully reflect a real situation exactly. However, it is still worth noting that it can be used as "a fair yardstick" in comparison with the performance of some other algorithms.
- Third, to assess the sensitivity of the adjustable parameters, a controlled experiment should be performed. This is almost impossible in a real situation.

The controllable parameters and performance measures in the experiments are as follows:

- Class imbalance ratio: A class imbalance ratio is the proportion of the 'Good' wafers and the total wafers. Figure 3 represents a yield distribution at a certain inspection step. This inspection step has 4 tools, and the yield distribution is shown as a raw dotted scatter plot with a box plot In this figure, the class imbalance ratio is (ⓐ+ⓑ)/(ⓐ+ⓑ+ⓒ+ⓓ). We use a class imbalance ratios of 0.80, 0.85, 0.90, and 0.95 in each experiment to determine the sensitivity of the algorithm's performance with the class imbalance ratio.
- Abnormality: An abnormality is the proportion of

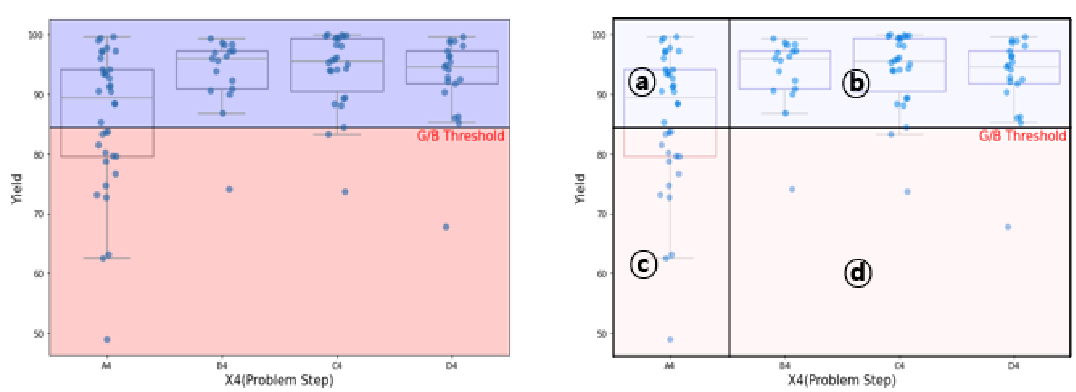


Fig. 3. Box plot of the candidate contrast sets (an attribute – value pair).

$support(\mathrm{X}, '\mathrm{Bad}')$ and $support('\mathrm{Bad}')$. In Figure 3, an abnormality is ⓒ/(ⓒ+ⓓ). We use abnormalities of 0.25, 0.5, 0.75, and 1.0 in each experiment to determine the sensitivity of the algorithm's performance with the abnormalities.

- Missing ratio: A missing ratio is the ratio of the missing fields to the total number of records in the history log.
- The number of steps: This is the column size of the history log table.
- Hit ratio: A hit ratio is defined as the success number divided by 100 random trials. Success means that the algorithm identifies the intentionally devised excursion tool as the most suspicious tool, i.e., top-1.
- Sample size: A sample size is the number of wafers. We assume that the number of wafers in this experiment is 100.

We compared the performance of a regression method, classification methods, and contrast set mining methods. Table IV shows the compared methods. Because our algorithm is based on the STUCCO algorithm, we classified our algorithm as contrast set mining. Additionally, we test TAR2 as a treatment learning method. In the classification methods, we use an RF, CART, and SVM. Finally, in the regression methods, we use only PLS-VIP.

TABLE IV
METHODS COMPARING THE PERFORMANCE OF REGRESSION, CLASSIFICATION, AND CONTRAST SET MINING

	Regression	Classification	Contrast set mining /Treatment Learning
Methods	PLS-VIP	RF, CART, SVM	TAR2, STUCCO, Our algorithm

The continuous yields are generated by an exponential distribution using a certain parameter λ. These values are binned by a certain threshold percentile determined by a given class imbalance ratio. Each column value of a chunk can have 3~5 unique strings using a uniform distribution. Only the suspicious chunk has 4 unique strings because we want to test abnormalities. For the sake of clarity, we assume that each step chunk can have five manufacturing steps and one inspection step.

B. Experiment 1: Robustness to class imbalance

We test our algorithm with a regression method, classification methods, and contrast set mining methods. In Table V, we summarize the performance of the algorithms. In the experiment, we fixed the number of steps and missing ratio as 10 and 0, respectively. All results are the mean value of 100 random trials. All algorithms show performance deterioration as the class imbalance ratio increases. Additionally, performance deteriorates as abnormalities grow lower. However, our algorithm is significantly more robust than the other algorithms, such as PLS-VIP, RF, CART, SVM, STUCCO and TAR2, in all conditions. This result is identical to the reference of [11]. When the abnormality is given as low as 0.25, it shows a similar low performance among all algorithms. Because we made 4 unique strings in the suspicious chunk, it is a natural consequence. Note that this imitates an *in-control* process situation since we assigned four tools at the predefined target contrast set. However, our algorithm shows superiority to other algorithms as soon as an abnormality grows over 0.5.

C. Experiment 2: Robustness to the number of steps

The number of fabrication steps also affects the performance of CA algorithms. Table VI depicts the relationship between the number of fabrication steps and the performance of the algorithms. We evaluate the performance of the algorithms under the condition that the step size is 10, 100, 200 and 300. As the step size increases, the performance of all algorithms gradually worsens. However, the performance of our algorithm and the STUCCO algorithm is relatively more robust than the other algorithms, such as PLS-VIP, RF, CART, SVM, and TAR2, in all conditions. Comparing our algorithm and the STUCCO algorithm, our algorithm performs more robustly. The reason is that the meaningless inspection steps with the influence of the main steps are removed due to the MCF. The larger the step size and class imbalance ratio are, the larger the gap in performance due to the MCF. Additionally, even if the class imbalance ratio is changed in the same step size, the performance of our algorithm does not change much. This is important because the sample size of the bad group is flexible in real analysis.

TABLE V
EXPERIMENT OF ALGORITHM PERFORMANCE COMPARED TO CLASS IMBALANCE

ABN	.25								.5								.75								1.00							
CI	.8	.85	.9	.95	.8	.85	.9	.95	.8	.85	.9	.95	.8	.85	.9	.95	.8	.85	.9	.95	.8	.85	.9	.95	.8	.85	.9	.95	.8	.85	.9	.95
Ours	.15	.21	.15	.17	.38	.38	.34	.21	.59	.57	.60	.51	1.00	1.00	1.00	.98	.15	.21	.15	.17	.38	.38	.34	.21	.59	.57	.60	.51	1.00	1.00	1.00	.98
STU	.12	.17	.13	.15	.37	.35	.29	.19	.56	.51	.49	.31	.67	.63	.62	.52	.12	.17	.13	.15	.37	.35	.29	.19	.56	.51	.49	.31	.67	.63	.62	.52
RF	.05	.04	.06	.09	.16	.19	.19	.22	.41	.42	.41	.35	.61	.57	.62	.48	.05	.04	.06	.09	.16	.19	.19	.22	.41	.42	.41	.35	.61	.57	.62	.48
PLS	.09	.12	.11	.09	.32	.29	.24	.09	.51	.47	.41	.25	.64	.56	.53	.42	.09	.12	.11	.09	.32	.29	.24	.09	.51	.47	.41	.25	.64	.56	.53	.42
TAR2	.04	.05	.07	.05	.09	.09	.16	.05	.20	.16	.26	.17	.41	.30	.40	.44	.04	.05	.07	.05	.09	.09	.16	.05	.20	.16	.26	.17	.41	.30	.40	.44
SVM	.02	.03	.08	.06	.07	.13	.19	.14	.23	.23	.26	.30	.49	.40	.44	.52	.02	.03	.08	.06	.07	.13	.19	.14	.23	.23	.26	.30	.49	.40	.44	.52
DT	.07	.10	.04	.05	.23	.24	.19	.11	.50	.45	.41	.31	.65	.55	.56	.50	.07	.10	.04	.05	.23	.24	.19	.11	.50	.45	.41	.31	.65	.55	.56	.50

[ABN: Abnormality, CI: Class Imbalance, STU: STUCCO, DT: Decision Tree]

This article has been accepted for publication in IEEE Transactions on Semiconductor Manufacturing. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TSM.2022.3201654

TABLE VI
EXPERIMENT OF ALGORITHM PERFORMANCE COMPARED TO THE NUMBER OF FABRICATION STEPS

ABN	.25																.5															
CI	.8				.85				.9				.95				.8				.85				.9				.95			
SIZE	10	100	200	300	10	100	200	300	10	100	200	300	10	100	200	300	10	100	200	300	10	100	200	300	10	100	200	300	10	100	200	300
Ours	.15	.16	.07	.06	.21	.19	.05	.03	.15	.14	.08	.02	.17	.18	.03	.06	.38	.39	.17	.18	.38	.35	.12	.10	.34	.29	.21	.14	.21	.26	.17	.12
STU	.12	.15	.06	.05	.17	.19	.05	.03	.13	.11	.08	.02	.15	.18	.03	.06	.37	.38	.16	.17	.35	.32	.11	.10	.29	.25	.20	.12	.19	.20	.12	.12
RF	.05	.00	.00	.00	.04	.01	.00	.00	.06	.01	.00	.00	.09	.00	.00	.00	.16	.12	.01	.02	.19	.02	.03	.04	.19	.07	.01	.01	.22	.00	.01	.02
PLS	.09	.06	.20	.02	.12	.09	.09	.04	.11	.16	.05	.09	.09	.04	.06	.02	.32	.06	.26	.05	.29	.17	.11	.07	.24	.25	.07	.11	.09	.05	.10	.06
TAR2	.04	.08	.03	.02	.05	.10	.03	.04	.07	.11	.02	.02	.05	.07	.02	.03	.09	.10	.07	.04	.09	.11	.06	.05	.16	.19	.07	.04	.05	.15	.06	.07
SVM	.02	.16	.06	.02	.03	.17	.01	.11	.08	.23	.05	.05	.06	.19	.08	.11	.07	.16	.07	.04	.13	.20	.04	.14	.19	.31	.14	.12	.14	.36	.13	.17
DT	.07	.11	.03	.03	.10	.14	.06	.05	.04	.15	.07	.10	.05	.13	.08	.12	.23	.12	.06	.12	.24	.24	.09	.13	.19	.21	.12	.18	.11	.14	.10	.19

ABN	.75																1.00															
CI	.8				.85				.9				.95				.8				.85				.9				.95			
SIZE	10	100	200	300	10	100	200	300	10	100	200	300	10	100	200	300	10	100	200	300	10	100	200	300	10	100	200	300	10	100	200	300
Ours	.59	.45	.43	.42	.57	.46	.32	.35	.60	.53	.45	.34	.51	.45	.38	.32	1.00	1.00	1.00	1.00	1.00	1.00	.98	.99	1.00	.99	.95	.91	.98	.88	.42	.70
STU	.56	.43	.35	.38	.51	.43	.24	.30	.49	.45	.35	.27	.31	.32	.25	.32	.67	.53	.50	.52	.63	.56	.47	.52	.62	.53	.50	.28	.52	.48	.29	.53
RF	.41	.37	.04	.10	.42	.02	.09	.10	.41	.10	.02	.05	.35	.01	.02	.02	.61	.39	.12	.15	.57	.07	.13	.16	.62	.10	.09	.07	.48	.01	.02	.03
PLS	.51	.08	.30	.06	.47	.27	.16	.14	.41	.26	.09	.14	.25	.12	.14	.06	.64	.15	.34	.11	.56	.33	.23	.18	.53	.27	.14	.14	.42	.27	.14	.07
TAR2	.20	.12	.11	.07	.16	.11	.09	.11	.26	.24	.11	.12	.17	.19	.11	.08	.41	.14	.16	.11	.30	.19	.15	.14	.40	.35	.18	.13	.44	.22	.12	.15
SVM	.23	.19	.10	.07	.23	.26	.09	.17	.26	.39	.22	.17	.30	.43	.22	.34	.49	.21	.15	.11	.40	.32	.15	.18	.44	.39	.27	.17	.52	.44	.23	.39
DT	.50	.14	.12	.14	.45	.24	.15	.18	.41	.25	.15	.21	.31	.27	.17	.35	.65	.15	.16	.16	.55	.27	.22	.19	.56	.27	.23	.33	.50	.29	.17	.41

[ABN: Abnormality, CI: Class Imbalance, STU: STUCCO, DT: Decision Tree]

D. Experiment 3: Robustness to missing values

Table VII shows the relationship between the degree of missing values and the performance of the algorithms. In this experiment, the step size number was fixed at 10. We randomly select a cell in a wafer history log and erase its value. We set the percentage of missing cells to 10%, 30%, 50% and 70%. The performance of all algorithms gradually worsens as many missing data occur. Although there is a variation in the performance, our algorithm shows better performance than the others in general.

E. Case Study

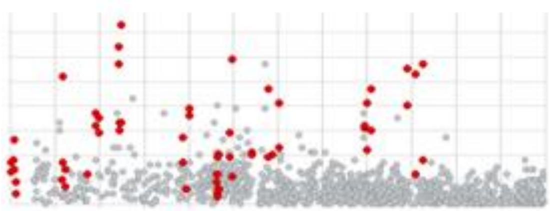
Figure 4 shows a real problem case of yield analysis in semiconductor fabrication. The root cause of the case was a

TABLE VII
EXPERIMENT OF ALGORITHM PERFORMANCE COMPARED TO THE NUMBER OF MISSING CELLS

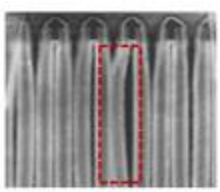
ABN	.25																.5															
CI	.8				.85				.9				.95				.8				.85				.9				.95			
MISS	10%	30%	50%	70%	10%	30%	50%	70%	10%	30%	50%	70%	10%	30%	50%	70%	10%	30%	50%	70%	10%	30%	50%	70%	10%	30%	50%	70%	10%	30%	50%	70%
Ours	.16	.48	.04	.15	.17	.00	.10	.22	.08	.06	.11	.08	.12	.05	.06	.07	.36	.60	.17	.20	.34	.18	.22	.35	.30	.19	.24	.16	.22	.21	.18	.23
STU	.13	.48	.04	.15	.13	.00	.10	.20	.11	.06	.11	.08	.10	.06	.09	.09	.30	.60	.15	.19	.32	.18	.21	.32	.23	.19	.19	.16	.22	.18	.18	.26
RF	.02	.02	.03	.08	.09	.00	.04	.20	.07	.01	.07	.07	.09	.02	.10	.09	.14	.19	.12	.11	.17	.18	.16	.25	.16	.05	.15	.19	.19	.09	.19	.21
PLS	.09	.47	.09	.24	.17	.00	.08	.13	.09	.10	.06	.10	.07	.00	.10	.15	.31	.57	.21	.28	.29	.12	.19	.18	.18	.20	.18	.22	.17	.13	.17	.28
TAR2	.02	.01	.06	.12	.13	.00	.04	.07	.07	.04	.08	.07	.01	.04	.09	.08	.09	.11	.14	.19	.23	.11	.12	.16	.14	.07	.17	.17	.06	.10	.17	.17
SVM	.04	.00	.04	.06	.20	.00	.05	.09	.05	.12	.05	.08	.05	.01	.09	.10	.12	.08	.15	.16	.31	.13	.16	.19	.12	.19	.09	.19	.12	.10	.16	.22
DT	.06	.01	.05	.14	.19	.00	.08	.07	.10	.00	.08	.08	.04	.02	.04	.02	.26	.13	.14	.19	.38	.12	.18	.22	.22	.03	.14	.18	.18	.13	.08	.03

ABN	.75																1.00															
CI	.8				.85				.9				.95				.8				.85				.9				.95			
MISS	10%	30%	50%	70%	10%	30%	50%	70%	10%	30%	50%	70%	10%	30%	50%	70%	10%	30%	50%	70%	10%	30%	50%	70%	10%	30%	50%	70%	10%	30%	50%	70%
Ours	.64	.76	.31	.32	.52	.43	.52	.55	.60	.49	.45	.29	.45	.45	.32	.28	.99	.95	.80	.49	1.00	.86	.80	.68	.97	.87	.73	.54	.50	.72	.53	.41
STU	.47	.69	.30	.31	.47	.34	.26	.41	.40	.37	.32	.29	.31	.32	.30	.31	.58	.73	.49	.41	.67	.46	.49	.46	.58	.49	.44	.45	.44	.51	.45	.39
RF	.30	.35	.27	.24	.29	.29	.19	.30	.32	.21	.30	.31	.33	.28	.24	.30	.52	.45	.41	.32	.46	.52	.36	.39	.51	.34	.40	.45	.34	.37	.33	.40
PLS	.52	.67	.21	.53	.38	.24	.35	.28	.33	.39	.32	.28	.26	.33	.29	.42	.68	.73	.47	.60	.56	.35	.56	.35	.54	.49	.40	.44	.27	.48	.43	.46
TAR2	.32	.18	.27	.41	.43	.21	.15	.26	.26	.18	.34	.30	.17	.20	.24	.23	.51	.31	.40	.48	.53	.36	.32	.30	.48	.38	.47	.42	.21	.33	.35	.33
SVM	.29	.28	.16	.28	.43	.22	.22	.23	.27	.37	.24	.27	.20	.31	.19	.27	.48	.43	.37	.37	.57	.36	.39	.29	.51	.52	.32	.42	.23	.43	.32	.33
DT	.42	.28	.39	.41	.44	.23	.34	.43	.36	.27	.29	.24	.23	.22	.10	.03	.55	.42	.51	.50	.57	.34	.54	.52	.49	.41	.35	.33	.25	.30	.22	.06

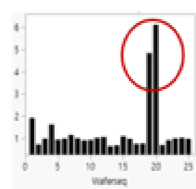
[ABN: Abnormality, CI: Class Imbalance, STU: STUCCO, DT: Decision Tree] bending phenomenon due to the disappearance of the polymer layer during inspection step A. Figure 4 (a) is a scatter plot for bridge failure in the wafer test. The red dots indicate bridge failure wafers of the same tendency. Figure 4 (b) shows the cutting plane of the fail wafers. In this image, two pillars are stuck together. This phenomenon is called the bending phenomenon. Due to the bending phenomenon, current leakage between the pillars was generated. The bridge failure related to current leakage was raised. Figure 4 (c) is a bar chart of the fail rate with the wafer sequence. In this figure, the failure is only generated in the 19th and 20th wafers of the lot. By checking the steps associated with this property, only inspection step A appears to be significant. As a result, the inspection points in a wafer affect the disappearance of the polymer layer while proceeding to tool B902 of step A. Therefore, we define that the problem chunk is tool B902 of step A.



(a) Scatter plot for distribution fail



(b) Image of cutting plane of problem wafer



(c) Fail rate with wafer sequence

Fig. 4. A case of yield analysis in semiconductor fabrication.

In this case, there are three difficulties: class imbalance, the number of steps, and missing values. Table VIII shows the log history of the case. The total number of wafers is 2,797. The good group's count is 2,743. The bad group's count is 54. The class imbalance ratio is very high at approximately 0.98. The number of inspection steps is 931. The number of manufacturing steps is 928. The number of target steps is 931 in algorithms other than ours because the algorithms target only the inspection step. On the other hand, in our algorithm, it is 1,859. Therefore, the number of features is much greater than that in the previous experiment in this paper. Finally, the average inspection ratio is only 2% to approximately 5% in semiconductor fabrication. Therefore, the missing ratio is 2% to approximately 5%. As can be seen in Table VIII, many wafers passed through the inspection steps.

We applied the algorithms to this case to compare their performance. Table IX is a comparison result table of the algorithms. The PLS-VIP and CART algorithms cannot create a model because there are approximately one thousand dimensions, which is too many. Therefore, the algorithms are excluded. Due to algorithmic properties, the RF only determines steps. Therefore, if it determines the problem step, it is considered to have determined the problem chunk in the case of the RF. Generally, most engineers tend to check the top 5 chunks of all suspicious chunks. Therefore, although an algorithm has more than 5 suspicious chunks, we display only the top 5 results.

In the SVM, there is no problem chunk in the top 5 suspicious chunks. On the other hand, there is the problem chunk in the results of our algorithm, the STUCCO, RF, and TAR2 algorithms. However, the problem chunk comes out as the third suspicious chunk in the RF algorithm and the fourth suspicious chunk in the STUCCO and TAR2 algorithms. In our algorithm, the problem chunk is detected as the first suspicious chunk. The difference between our algorithm and other algorithms is the consideration of the relationship between the inspection step and their manufacturing step. We compare to the bad group's count in the steps to determine the effect of the relation. Table X shows the bad wafer counts in the inspection steps and their

TABLE VIII

LOG HISTORY FOR A CASE OF YIELD ANALYSIS IN FABRICATION

Wafer-ID	…	Inspection Step A	Manufacturing Step A	…	Inspection Step C	Manufacturing Step C	…	Group
1	…	Null	'2405	…	Null	'1504'	…	'Good'
2	…	Null	'2405'	…	Null	'1504'	…	'Good'
3	…	'B902'	'2405'	…	'Null'	'1504'	…	'Bad'
4	…	'B902'	'0501'	…	'0202'	'1504'	…	'Bad'
…	…	…	…	…	…	'1504'	…	…
2797	…	'4001'	'2405'	…	Null	'1504'	…	'Good'

TABLE IX

EXPERIMENT OF ALGORITHM PERFORMANCE COMPARED TO REAL YIELD ANALYSIS IN FABRICATION

No.	Ours	STUCCO	RF	SVM	TAR2
1	A_EQ=B902	K_EQ=1902	B_EQ	P_EQ=0602	C_EQ=0202
2	K_EQ=1902	B_EQ=D202	K_EQ	S_EQ=1401	K_EQ=1902
3	B_EQ=D202	A_EQ=B902	G_EQ	C_EQ=0202	A_EQ=B902
4	None	None	A_EQ	D_EQ=0201	Q_EQ=E601
5	None	None	U_EQ	L_EQ=1902	Y_EQ=4301

TABLE X
COUNT OF THE BAD GROUP IN THE INSPECTION STEPS AND THE MANUFACTURING STEPS
(IN THE RESULTS OF THE EXPERIMENT OF ALGORITHM PERFORMANCE).

(A) INSPECTION STEP A

Inspection Tools	Manufacturing Tools	Wafer Count	Abnormality
4001	2405	5	.91
B902	0501	25	
B902	2405	27	

(B) INSPECTION STEP C

Inspection Tools	Manufacturing Tools	Wafer Count	Abnormality
Null	1504	23	.56
0202	1504	32	
0701	1504	2	

(C) INSPECTION STEP K

Inspection Tools	Manufacturing Tools	Wafer Count	Abnormality
1902	E506	10	1.00
1902	E509	47	

(D) INSPECTION STEP B

Inspection Tools	Manufacturing Tools	Wafer Count	Abnormality
D202	2108	13	1.00
D202	2110	44	

(E) INSPECTION STEP G

Inspection Tools	Manufacturing Tools	Wafer Count	Abnormality
1902	0702	57	1.00

manufacturing steps. Because inspection steps B, C, G, and K are higher ranks than inspection step A in the results of the STUCCO, RF and TAR2 algorithms, the inspection steps are only compared to inspection step A. In the case of inspection step G, all bad wafers are processed by only one inspection tool. Therefore, the step seems to be a real problem chunk. However, if the manufacturing step is considered, it cannot be the problem chunk because the inspection tool and the manufacturing step's tools have almost the same bad wafer rate. This means that it is difficult to know whether the root cause is the inspection step or manufacturing step. In inspection step C, there are 23 bad wafers that did not proceed to the inspection step. The number of these wafers is almost half of the total bad wafers. The missing data are removed in the TAR2 algorithm because it cannot make contrast set to them. This means that only half of the total bad wafers are used in the TAR2 algorithm. Therefore, it is difficult to conclude that inspection step C is the root cause.

In the cases of K and B, they also exist in the results of our algorithm. In contrast to the STUCCO algorithm, they are secondary suspicious chunks to the problem chunk in our algorithm. The reason is that the value of the MCF is lower than the problem chunk. In inspection steps K and B, the distribution of the manufacturing tools is similar to the distribution of the manufacturing tools in inspection step A. However, the MCF of inspection step A is higher than others because the bad wafers' count of manufacturing tools in inspection step A is more even. Therefore, among the three suspicious chunks, inspection step A is the most suspicious step. The abnormality of inspection step A is 0.91.

IV. CONCLUSION

This paper has discussed the problem of finding an inspection tool excursion in semiconductor fabrication. Only a limited volume of the wafers are passed through the inspection tools. Additionally, class imbalance and missing fields in a history log table make traditional commonality analysis difficult for inspection tools. To cope with these difficulties, we propose a new algorithm to detect excursion of semiconductor inspection tools. We have shown that our algorithm is superior to other commonality algorithms based on a typical regression or classification model. We have shown good performance in the real yield analysis case. Specifically, the MCF, the relationship degree between manufacturing and inspection, has contributed significantly to reducing the number of suspicious chunks and determining a problem chunk. It is expected to increase the accuracy of the analysis and reduce the time required to determine the root cause of yield loss and the workload of yield analysis engineers. Actually, when we used the proposed algorithm as a specialized analytical algorithm for inspection in our FAB, the accuracy and persistency are superior to those of a legacy algorithm.

However, our experiment targets only the inspection steps and tools. Most steps in semiconductor fabrication are manufacturing steps. As the number of the manufacturing steps increase, the number of combinations between steps and tools that algorithms should explore also increases resulting in an increase in the amount of computation and the search time. Therefore, to apply this algorithm to a manufacturing environment, it seems that it should be targeted to a specific chunk.

In addition to the inspection steps, there are many dedicated steps among the manufacturing steps themselves. In the semiconductor manufacturing environment, it proceeds with a combination of specific tools in the forward and backward steps to maintain high yield. A step with this combination of tools is called the dedicated step. In most such cases, it is difficult to determine which process is a clear root cause. The proposed algorithm can increase accuracy by using MCF to eliminate noise for combinations of tools. Therefore, it seems that our algorithm can be applicable to the dedicated steps between the manufacturing steps. However, because the dedicated steps between the manufacturing steps is more complicated, many discussions with engineers are required.

REFERENCE

[1] Liam Y. Hsieh, Tsung-ju Hsieh, "A Throughput Management System for Semiconductor Wafer Fabrication Facilities: Design, Systems and Implementation", Processes, Vol.6, No.16, 2018.

[2] George Kong, "Tool Commonality Analysis for Yield Enhancement", 13th Annual IEEE/SEMI Advanced Semiconductor Manufacturing Conference, 30 April~2 May, 2002, pp.202~205.

[3] H.-A. Kao, Y.-S. Hsieh, C.-H. Chen and J. Lee, "Quality prediction modeling for multistage manufacturing based on classification and association rule mining," Proc. MATEC Web Conf., vol. 123, pp. 29, Sep. 2017.

[4] H. Barkia, X. Boucher, R. Le Riche, P. Beaune, M. A. Girard and D. Rozier, "Semiconductor yield loss' causes identification: A data mining approach," 2013 IEEE International Conference on Industrial Engineering and Engineering Management, 2013, pp. 843-847.

[5] T. Heo, Y. Kim and C. O. Kim, "A Modified Lasso Model for Yield Analysis Considering the Interaction Effect in a Multistage Manufacturing Line," in IEEE Transactions on Semiconductor Manufacturing, vol. 35, no. 1, pp. 32-39, Feb. 2022.

[6] M. Y. Lee, Y. J. Choi, G. T. Lee, J. Choi and C. O. Kim, "Attention mechanism-based root cause analysis for semiconductor yield enhancement considering the order of manufacturing processes," in IEEE Transactions on Semiconductor Manufacturing.

[7] Sholom M. Weiss, Robert J. Baseman, Fateh Tipu , Christopher N. Collins, William A. Davies, Raminderpal Singh, John W. Hopkins, "Rule-based data mining for yield improvement in semiconductor manufacturing", Applied Intelligence, Vol.33, pp.318 - 329, 2010.

[8] A. Nutsch, R. Oechsner, U. Schoepka and L. Pfitzner, "Yield model for estimation of yield impact of semiconductor manufacturing equipment," 2010 International Symposium on Semiconductor Manufacturing (ISSM), 2010, pp. 1-4.

[9] Chen-Fu Chien, Chiao-Wen Liu & Shih-Chung Chuang, "Analysing semiconductor manufacturing big data for root cause detection of excursion for yield enhancement," International Journal of Production Research, 2017, 5095-5107

[10] Weiss, S.M., Baseman, R.J., Tipu, F. et al., "Rule-based data mining for yield improvement in semiconductor manufacturing," Appl Intell 33, 2010, 318–329.

[11] Jiuyong Li, Ada Wai-chee Fu, Hongxing He, Jie Chen, Huidong Jin, Damien McAullay, Graham Williams, Ross Sparks, Chris Kelman, "Mining Risk Patterns in Medical Data", KDD'05 August 21 - 24, 2005, Chicago, Illinois, USA, pp.770~775.

[12] Rakesh Agrawal, Ramakrishnan Srikant, "Fast Algorithms for Mining Association Rules", VLDB '94: Proceedings of the 20th International Conference on Very Large Data Bases, September, 1994, pp.487-499.

[13] Han, J. , Pei, J. , & Yin, Y. "Mining frequent patterns without candidate generation". In Proc. ACM-SIGMOD Int. Conf. Management of Data (SIGMOD '96), Page 205-216, 2000.

[14] M. Zaki, S. Parthasarathy, M. Ogihara, and W. Li., "New Algorithms for Fast Discovery of Association Rules. Proc. 3rd Int. Conf. on Knowledge Discovery and Data Mining (KDD'97), 283–296. AAAI Press, Menlo Park, CA, USA 1997.

[15] Jong Soo Park, Ming-Syan Chen and Philip S. Yu, "An Effective Hash-Based Algorithm for Mining Association Rules", In SIGMOD'95

[16] Jian Pei, Jiawei Han, and Runying Mao, "CLOSET: An Efficient Algorithm for Mining Frequent Closed Itemsets", In Proceedings of the 2000 ACM SIGMOD Workshop on Research Issues in Data Mining and Knowledge Discovery. Dallas, Texas, USA, pp. 21–30.

[17] S.D. Bay, M.J. Pazzani, "Detecting group differences: mining contrast sets", Data Mining and Knowledge Discovery, 5 (3) (2001), pp. 213-246.

[18] Y. Hu (2003). Treatment learning: Implementation and application (Master's thesis). Department of Electrical Engineering, University of British Columbia.

[19] Tim Menzies, Ying Hu, "Just Enough Learning (of Association Rules): The TAR2 "Treatment" Learner", Artificial Intelligence Review (2006), Vol.25, pp.211-229

[20] R. -H. Chen and C. -M. Fan, "Treatment of missing values for association rule-based tool commonality analysis in semiconductor manufacturing," 2012 IEEE International Conference on Automation Science and Engineering (CASE), 2012, pp. 886-891, doi: 10.1109/CoASE.2012.6386510.

[21] Choi Doo Won, Sim Hyun Sik, Kim Chang Ouk, "Discovery of Process Equipments Causing Product Faults in a Nano-Scale Manufacturing Line", Proceedings of Conference of the Korean Institute of Industrial Engineering, May. 2013, pp.1185~1192.