

## **MARKOV-CHAIN BASED MISSING VALUE ESTIMATION METHOD FOR TOOL COMMONALITY ANALYSIS IN SEMICONDUCTOR MANUFACTURING**

Rong-Huei Chen

National Taiwan University  
No.1, Sec. 4, Roosevelt Road

Chih-Min Fan

Yuan Ze University  
135, Yuan-Tung Road

### **ABSTRACT**

Association rule-based tool commonality analysis (ARBTC) is an effective approach to identifying tool excursions for yield enhancement in semiconductor manufacturing. However, missing values which frequently occurred will lead to high rates of false positive and false negative. Incorrect identification of root cause of yield loss will lose engineer's trust on TCA and delay the process improvement opportunity. In this paper, we proposed a Markov-chain based Missing Value Estimation (MCBMVE) method to improve the effectiveness of ARBTC, and demonstrate and explain why traditional methods dealing with missing values for association rules cannot solve the problem. Comparing with traditional methods, the real case study shows that MCBMVE is more accurate in recovering missing values so as to improve the identification accuracy..

### **1 INTRODUCTION**

In semiconductor manufacturing, there are hundreds of processing steps with multiple tools at most steps. Any tool excursion in a processing step may result in product yield loss and decrease manufacturing profit. Though various in-line inspections established to monitor individual tools, none of them is guaranteed to successfully isolate all the root causes of product yield loss [1]. Some tool excursions related to yield loss cannot be identified by in-line inspections but are only observable through end-of-line tests such as electrical test (E-test) and circuit probing (CP) test. Once an end-of-line yield loss event is detected, how to effectively identify a specific tool excursion from hundreds of processing steps as the root cause is a permanent challenge to a modern semiconductor manufacturing fab.

Tool commonality analysis (TCA) is an emerging topic for the effective identification of tool excursions using end-of-line yield data. Given a yield loss event with affected wafer yield and associated tool usage data, TCA iteratively conducts statistical hypotheses on individual equipment tools in production line and pinpoints which tool causes the wafer yield loss. There are two types of wafer yield input to TCA, continuous and Bernoulli. The continuous wafer yield is directly counted by the die yield on the wafer, where as the Bernoulli wafer yield is calculated by comparing the spatial signature of the die yield on the wafer to the pattern associated with the yield loss event. The TCA for continuous wafer yield adopts traditional statistical techniques such as ANOVA or contingency tables to search for fab tool commonalities. As for the Bernoulli wafer yield, the association rule method is adopted [2] to the detection and discovery of fab tool commonality for wafers with spatial signatures.

The soundness of TCA has a high impact on the effectiveness of product yield diagnosis. Unfortunately, the TCA techniques applying traditional statistical methods often result in high rates of false positive and false negative in yield analysis [5], requiring much time of engineers to review and validate commonality results (loop). Incorrect identification of root cause of yield loss not only loses engineer's trust on TCA, but also delays the process improvement opportunity. Many investigators try to solve the

problem of high false positive and high false negative rates in TCA [5], but they didn't address the impact of missing values.

In our research, we focus on studying the treatment of missing values for the association rule-based TCA with Bernoulli end-of-line input. We found that missing values also lead to high rates of false positive and false negative in association rule-based TCA. However, no literature of association rule-based TCA has considered the missing values. Furthermore, we also found that traditional methods dealing with missing values for association rule such as RAR [3] and MVC [4] are not suitable for TCA applications.

To cope with the problem of missing value, we adopt the Markov based probability model to evaluate the probability of missing values passing through which tool and in which time period in previous research [6]. In this paper, we compare with RAR and MVC method. First, we calculate conditional support and confidence pair with probabilities. Finally, we show our Markov based simulation method can decrease false positive and false negative rates in TCA effectively.

## 2 METHODOLOGY SURVEY FOR TOOL COMMONALITY

The semiconductor manufacturing processes include several hundred of processing steps with multiple tools at most steps. The total number of tools across all of the steps typically exceeds 1000. Each lot includes 25 wafers and is processed by a single tool at each processing step. Besides, each wafer can have several thousands of die. Tool trajectories which are the sequence of tools at each step that processes a lot are determined by a scheduling algorithm. We define an error which is a tool that processes lots differently enough from other tools at the same step to impact performance of yield.

Though various in-line inspections established to monitor individual tools, none of them is guaranteed to successfully isolate all the root causes of product yield loss [1]. Some tool excursions related to yield loss cannot be identified by in-line inspections but are only observable through end-of-line tests such as electrical test (E-test) and circuit probing (CP) test. Once an end-of-line yield loss event is detected, engineers face the challenge of locating steps from hundreds of processing steps with yield losing with little data and many possibilities.

The yield analysis flow includes two parts, wafer pattern recognition and tool commonality analysis. In semiconductor manufacturing, current methodology for the detection and discovery of good/bad wafer is a manual process by checking the special patterns (a.k.a. spatial signatures) called wafer pattern recognition. Typically, wafers maps are reviewed by engineers. A score indicating the degree to which a wafer demonstrates the pattern is calculated. The wafer pattern recognition exist testing error because of manual process by engineers.

Discovering which factory tool is causing the problem is the ultimate goal of tool commonality analysis (TCA) using end-of-line yield data. Given a yield loss event with affected wafer yield and associated tool usage data, TCA iteratively conducts statistical hypotheses on individual equipment tools in production line and pinpoints which tool causes the wafer yield loss. Unfortunately, the TCA techniques applying traditional statistical methods often result in high rates false positive and false negative in yield analysis [5], requiring much time of engineers to review and validate commonality results (loop). Incorrect identification of root cause of yield loss not only loses engineer's trust on TCA but also delays the process improvement opportunity. To discuss TCA problem in semiconductor manufacturing, we first define some notations as following:

Table 1: Notations

$i$	Operation id
$j$	Tool id
$k$	Time period
$N_B$	Number of bad wafers
$N_{Bi}$	Number of bad wafers passing through operation $i$

$N_{Bi}^{(O)}$	Number of observed bad wafers passing through tool i
$N_{Bi}^{(M)}$	Number of missing bad wafers passing through operation i
$N_{ijk}$	Number of wafers passing through operation i, tool j and time period k
$N_{Bij}$	Number of bad wafers passing through operation i and tool j
$N_{Bijk}$	Number of bad wafers passing through operation i, tool j and time period k
$N_{ijk}^{(O)}$	Number of observed wafers passing through operation i, tool j and time period k
$N_{ijk}^{(M)}$	Number of missing wafers passing through operation i, tool j and time period k
$N_{Bijk}^{(O)}$	Number of bad observed wafers passing through operation i, tool j and time period k
$N_{Bijk}^{(M)}$	Number of bad missing wafers passing through operation i, tool j and time period k
$S_j$	Conditional support of tool j
$C_j$	Confidence of tool j
$N_{Gijk}^{(O)}$	Number of good observed wafers passing through operation i, tool j and time period k
$N_{Gijk}^{(M)}$	Number of good missing wafers passing through operation i, tool j and time period k

## 2.1 Association Rules

In this paper, we adopt association rules for tool commonality analysis with Bernoulli end-of-line input; the key idea is to efficiently search for the commonality data and look for fab tools and time periods where many of the affected wafers were processed and where only affected wafers were processed. we consider records of wafers which include passing through operation i, and tool j at time period k. we use association rule algorithm to calculate basic statistics of fab tool usages consisting of conditional support of tool j and confidence of tool j as following:

$$S_j = \frac{N_{Bij}}{N_B} \tag{1}$$

$$C_i = \frac{N_{Bijk}}{N_{ijk}} \tag{2}$$

Accordingly, the conditional support of root cause is necessarily equal to 1. However, because of the false identification in wafer pattern recognition, the conditional support (C.S.) cannot be equal to 1, but still it must be close to 1. For the necessary condition, we will define a minimal conditional support (in our scenario, we set min C.S = 0.6) and minimal confidence (in our scenario, we set min C = 0.6) to screen the tool if it is not root cause.

In addition, when C.S. is the same, the higher confidence value is the more suspected root cause. If passing the min conditional support and confidence thresholds, we rank the remaining rules based upon their distance from the Peroto frontier on the conditional support-confidence plane and generate a ranking table. Each point at the line is the non-dominated solution which has the highest rank. Furthermore, we rank other points by the distance to the Peroto frontier. For example, the left side of Fig.1 shows that points A, B, C, D and E are on the Peroto frontier, and they are the non-dominated solutions. The point F is the latest rank because it is not on the Peroto frontier. In addition, the point C is the highest rank because of the longest distance to the red line (right side of Fig. 1).

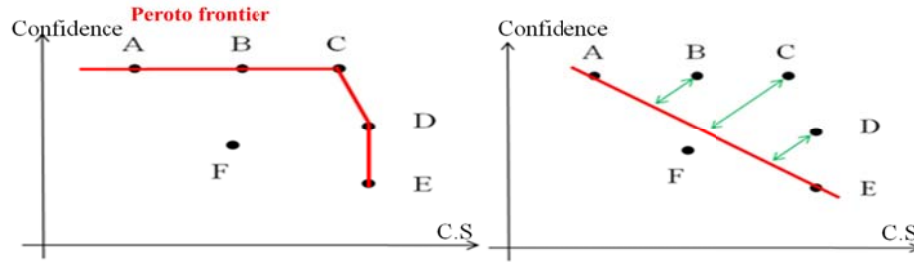


Figure 1: Ranking Rules on Peroto frontier

### 2.2 Association Rules with missing Value

With missing values, the association rules of conditional support of tool j and confidence of tool j are calculated as following:

$$S_j^{direct} = \frac{N_{Bij}^{(O)} + N_{Bij}^{(M)}}{N_B} = \frac{N_{Bij}^{(O)} + N_{Bij}^{(M)}}{N_B} = \frac{N_{Bij}^{(O)} + 0}{N_B} = \frac{N_{Bij}^{(O)}}{N_B} \tag{3}$$

$$C_j^{direct} = \frac{N_{Bijk}}{N_{ijk}} = \frac{N_{Bijk}^{(O)} + N_{Bijk}^{(M)}}{N_{ijk}^{(O)} + N_{ijk}^{(M)}} = \frac{N_{Bijk}^{(O)} + 0}{N_{ijk}^{(O)} + 0} = \frac{N_{Bijk}^{(O)}}{N_{ijk}^{(O)}} \tag{4}$$

We can see that the conditional support and confidence calculating with missing values will be lower than the calculating without missing values. Also, it's possible that the conditional support and confidence value is lower than real situation. It is not desirable because the conditional support and confidence are lower than the conditional support threshold significantly. The tool which is a root cause would be screened out easily. This phenomenon will lead to false identification.

### 2.3 Robust Association Rule to deal with missing Value

To deal with missing values, the Robust Association Rule (RAR) approach is one of the popular approaches used in association rules. The key idea of the Robust Association Rule (RAR) approach [3] is using the association rules method to partially discard the missing values. Based on this concept, the conditional support & confidence values are calculated by the RAR method as following:

$$S_j^{RAR} = \frac{N_{Bij}^{(O)}}{N_B - N_{Bij}^{(M)}} \tag{5}$$

$$C_j^{RAR} = \frac{N_{Bijk}^{(O)} + N_{Bijk}^{(M)}}{N_{ijk}^{(O)} + N_{ijk}^{(M)}} = \frac{N_{Bijk}^{(O)} + 0}{N_{ijk}^{(O)} + 0} = \frac{N_{Bijk}^{(O)}}{N_{ijk}^{(O)}} \tag{6}$$

In TCA, the RAR method can obtain the new conditional support values, but cannot adjust the confidence value. We can see that the conditional support value by RAR calculating will be higher than by association rules calculating. Also, it's possible that the conditional support value is higher than real situation. It's not good because the conditional supports are higher than the conditional support threshold significantly. The tool which is not a root cause would be identified as a root cause easily. The phenomenon will lead to false identification.

### 2.4 Missing Values Completion to deal with missing Value

To deal with missing values, the Missing Values Completion (MVC) approach is another popular approach used in association rules. The key idea of the Missing Values Completion (MVC) approach [3] is to apply association rules to fill in missing values. In the MVC method, only association rules with a high confidence value (more than 95%) can be used to deal with the missing value problem. Thus, the conditional support and confidence values are calculated by the MVC method as following:

$$S_j^{MVC} = \frac{N_{Bij}^{(O)} + N_{Bij}^{(M,MVC)}}{N_B} \tag{7}$$

,where  $N_{Bij}^{(M,MVC)}$  is the evaluation value of missing bad wafer which pass thorough tool j and operation i by the MVC method

$$C_j^{MVC} = \frac{N_{Bijk}}{N_{ijk}} = \frac{N_{Bijk}^{(O)} + N_{Bijk}^{(M,MVC)}}{N_{ijk}^{(O)} + N_{ijk}^{(M,MVC)}} \tag{8}$$

In TCA, the MVC method is not always useful, because the confidence value is always less than 95%. Also, even if the MVC method can obtain the new conditional support values, the conditional support calculation via MVC becomes higher than that without missing value consideration. The tool which is not a root cause would be identified as a root cause easily. Also, the conditional support and confidence value varies depending on the minimal confidence threshold value. Many tools often got the lower confidence without missing value consideration

### 3 MARKOV CHAIN BASED MISSING VALUE ESTIMATION (MCBMVE) ALGORITHM

The main concept is to model whether the wafer passes the machine as Bernoulli distribution, and then based on the observed data, we can estimate whether the wafer with missing values passed the machine. Since the ratio of semiconductor wafer passing the machine should not very much, we assume the missing value follows the same Bernoulli distribution. Because error must be present following this assumption, we deal with the effect of error through iteration. Therefore, we adopt the Markov model to deal with missing values, and the core is state transition.

Based on above reasons, we implement a methodology based on the Markov chain model using traditional association rules to deal with the missing value problem. the inputs of algorithm are tool-time usage and good/bad information. If exist missing value in tool-time usage data, then evaluate the tool-time usage in step 1. In step 2, the algorithm recalculates conditional support, which is the percent of bad wafers captured by the rule. By requiring a minimal conditional support, many tools and time periods are removed. After calculating confidence, we can further filter the number of rules needed to be ranked by requiring a minimum confidence. If passing the min conditional support and confidence thresholds, then we rank the passing rules based on their distance to the Peroto frontier on the conditional support-confidence plane and generate a ranking table.

#### Step 1 tool-time usage evaluation

The goal of step 1 is to evaluate tool usage for improvement of conditional support and evaluate tool-time usage to improve confidence. To evaluate tool usage, we first consider operation i and tool j. Suppose tool  $j^*$  in operation i is the root cause, the number of missing bad wafers passing through tool  $j^*$  in operation i is denoted as  $N_{Bij^*}^{(M)}$ , and it's satisfied following the equation:  $0 \leq N_{Bij^*}^{(M)} \leq N_{Bi}^{(M)}$ . We adopt the Markov Chain model to evaluate the missing bad wafers of operation i & tool j.

We create a Markov chain as follows: We have a set of states,  $S = \{s_1, s_3, s_2 \dots s_r\}$ . In our model, states are defined as the number of missing bad wafers passing through tool j in operation i ( $N_{Bij}^{(M)}$ ), and the all the states are depicted as following:

$$S_r : N_{Bij^*}^{(M)}, r \in N, 0 \leq r \leq N_{Bi}^{(M)} \tag{9}$$

The process starts in one of these states and moves successively from one state to another. Each move is called a step. If the chain is currently in state  $s_m$ , then it moves to state  $s_n$  at the next step with a probability denoted by  $p_{mn}$ , and this probability does not depend upon which states the chain was in before the current state. Thus, we define transition probability( $p_{mn}$ ) which is the transition probability from state m at iteration t to state n at iteration t+1. Based on tool time information, transition probability ( $p_{mn}$ ) can be calculated as the following equation:

$$P_{mn} = C_n^{N_{Bi}^{(M)}} (\hat{p}_{Bij^*})^{(n)} (1 - \hat{p}_{Bij^*})^{(N_{Bi}^{(M)} - n)} \tag{10}$$

where  $\hat{p}_{Bij^*} = \frac{N_{Bij^*}^{(O)} + m}{N_{Bi}^{(O)} + N_{Bi}^{(M)}}$ , and m is the state at iteration t, n is the state at iteration t+1.

We now consider the long-term iteration of a Markov chain when it starts in a state chosen by a probability distribution on the set of states, which we will call a probability vector. If u is a probability vector which represents the initial state of a Markov chain, then we think of the t-th component of u as representing the probability that the chain starts in state  $s_m$ . Let P be the transition matrix of a Markov chain, and let u be the probability vector which represents the starting distribution. Then the probability that the chain is in state  $s_m$  after n steps is the ith entry in the vector

$$u^{(n)} = uP^n \tag{11}$$

In this model, the initial states of transition matrix can be inferred by initial  $\hat{p}_{Bij^*}$ , and the convergence conditions is  $\Pr(\text{state}^t) = \Pr(\text{state}^{t+1})$ . we use the model to estimate the missing value of bad wafer in operation j and tool j, while the states and conditional support follow the probability distribution. As the same, we can use the same concept to evaluate tool-time usage.

**Step 2 recalculate the conditional support and confidence**

And then we consider the missing value, the number of bad wafer equal to number of bad wafer passing through operation j which includes observable and missing, and the statistics of conditional support of tool j and confidence of tool j are revised as following:

$$N_B = N_{Bi} = N_{Bi}^{(O)} + N_{Bi}^{(M)} = \sum_j N_{Bij}^{(O)} + \sum_j N_{Bij}^{(M)} \tag{12}$$

$$S_j = \frac{N_{Bij}}{N_B} = \frac{N_{Bij}^{(O)} + N_{Bij}^{(M)}}{N_B} \tag{13}$$

$$C_i = \frac{N_{Bijk}}{N_{ijk}} = \frac{N_{Bijk}^{(O)} + N_{Bijk}^{(M)}}{N_{ijk}^{(O)} + N_{ijk}^{(M)}} \tag{14}$$

**4 SIMULATION EXAMPLE TO COMPARE THE RESULT BETWEEN METHODS**

The objective of our simulation study is to demonstrate the effect between different methods in ARBTCA. We use a simple case which consists of 20 wafers and 2 operations including 2 tools per operation to demonstrate the result between different algorithms.

**4.1 Direct Calculation**

In this case, the root cause is Tool A1, and red word marks missing values as shown in Fig.2. The missing values include tool wafers passing through and the time period in which wafers passed through.

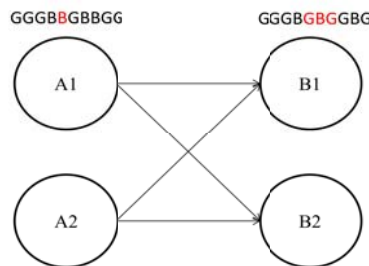


Figure 2: Simple Example for comarison between methods

Without missing value consideration, the conditional support & confidence values are directly calcu-

lated as following:

$$S_{A1}^{direct} = \frac{N_{Bij}^{(O)}}{N_B} = \frac{3}{6} = 0.5 \tag{15}$$

$$C_{A1}^{direct} = \frac{N_{Bijk}^{(O)} + N_{Bijk}^{(M)}}{N_{ijk}^{(O)} + N_{ijk}^{(M)}} = \frac{N_{Bijk}^{(O)} + 0}{N_{ijk}^{(O)} + 0} = \frac{3}{4} = 0.75 \tag{16}$$

Besides tool A1, we use another tool which is not a root cause as example. We consider the data set in tool B1, which consists of 20 wafers with 3 missing. The missing values include tool wafers passing through and the time period in which wafers passed through. Without considering missing values, the conditional support & confidence values of tool B1 by a direct method as following:

$$S_{B1}^{direct} = \frac{N_{Bij}^{(O)}}{N_B} = \frac{2}{6} = 0.33 \tag{17}$$

$$C_{B1}^{direct} = \frac{N_{Bijk}^{(O)} + N_{Bijk}^{(M)}}{N_{ijk}^{(O)} + N_{ijk}^{(M)}} = \frac{N_{Bijk}^{(O)} + 0}{N_{ijk}^{(O)} + 0} = \frac{2}{3} = 0.66 \tag{18}$$

We can see that Both A1 and B1 will be screen out because the conditional support value is less than the minimal conditional support.

#### 4.2 Calculation by RAR and MVC

We compare the results of our method with two existing methods to deal with missing values, the RAR approach and the MVC approach.

The result of tool A1 & B1 are shown in Table 2. We can see that the RAR method can obtain the new conditional support values, but cannot adjust the confidence value. Besides, the conditional support of B1 become higher than conditional support of A1. It is not desirable because the conditional supports of tool B1 is higher than the conditional support threshold and root cause tool significantly.

The same phenomenon is found in the MVC method. We can see the MVC method can obtain the new conditional support values, but the conditional support of B1 becomes higher than tool A1. MVC is not suitable for dealing with missing values in this case. In root cause diagnosis, both RAR method & MVC methods present high risk of false identification. For example, tool B1 is not the root cause, but the conditional support and confidence values are increased highly through RAR & MVC methods. It will cause a higher rank than the real root cause.

Table 2: Result of RAR, MVC method of tool A1& B1

$S_{A1}^{RAR} = \frac{N_{Bij}^{(O)}}{N_B} = \frac{3}{5} = 0.6$	$S_{B1}^{RAR} = \frac{N_{Bij}^{(O)}}{N_B} = \frac{2}{6-3} = 0.66$
$S_{A1}^{MVC} = \frac{N_{Bij}^{(O)}}{N_B} = \frac{3+1}{6} = 0.66$	$S_{B1}^{MVC} = \frac{N_{Bij}^{(O)}}{N_B} = \frac{3+3}{6} = 1$
$C_{A1}^{RAR} = \frac{N_{Bijk}^{(O)} + N_{Bijk}^{(M)}}{N_{ijk}^{(O)} + N_{ijk}^{(M)}} = \frac{N_{Bijk}^{(O)} + 0}{N_{ijk}^{(O)} + 0} = \frac{3}{4} = 0.75$	$C_{B1}^{RAR} = \frac{N_{Bijk}^{(O)} + N_{Bijk}^{(M)}}{N_{ijk}^{(O)} + N_{ijk}^{(M)}} = \frac{N_{Bijk}^{(O)} + 0}{N_{ijk}^{(O)} + 0} = \frac{2}{3} = 0.66$
$C_{A1}^{MVC} = \frac{N_{Bijk}^{(O)} + N_{Bijk}^{(M)}}{N_{ijk}^{(O)} + N_{ijk}^{(M)}} = \frac{N_{Bijk}^{(O)} + 1}{N_{ijk}^{(O)} + 1} = \frac{4}{5} = 0.8$	$C_{B1}^{MVC} = \frac{N_{Bijk}^{(O)} + N_{Bijk}^{(M)}}{N_{ijk}^{(O)} + N_{ijk}^{(M)}} = \frac{N_{Bijk}^{(O)} + 3}{N_{ijk}^{(O)} + 3} = \frac{5}{6} = 0.83$

#### 4.3 Calculation by MCBMVE

We go through the MCBMVE algorithm to complete the missing value of tool B1 100 times, and then recalculate conditional support and confidence as following:

$$S_{A1}^{MCBMVC,ave} = \frac{N_{Bij}^{(O)}}{N_B} = \frac{3+1}{6} = 0.6 \tag{19}$$

$$C_{A1}^{MCBMVC,ave} = \frac{N_{Bijk}^{(O)} + N_{Bijk}^{(M)}}{N_{ijk}^{(O)} + N_{ijk}^{(M)}} = \frac{3+1}{4+1} = \frac{4}{5} = 0.80 \tag{20}$$

In addition, the re-calculated conditional support & confidence values of tool H1 and tool H2 by MCBMVE method, and the expected value of conditional support & confidence are calculated as following:

$$S_{B1}^{MCBMVC} = \frac{N_{Bij}^{(O)}}{N_B} = \frac{3+1}{6} = 0.495 \tag{21}$$

$$C_{B1}^{MCBMVC} = \frac{N_{Bijk}^{(O)} + N_{Bijk}^{(M)}}{N_{ijk}^{(O)} + N_{ijk}^{(M)}} = \frac{N_{Bijk}^{(O)} + 0}{N_{ijk}^{(O)} + 0} = \frac{3}{6} = 0.5 \tag{22}$$

We found that the conditional support-confidence pair of tool A1 has the highest conditional support (1) and confidence value (1). It leads the ranking of tool A1 to rank 1.

We also found that the tool B1 will be screened out by the minimal conditional support. After using SBMVC algorithm to complete missing values, the tool that is not the root cause may not have largely changed in conditional support and confidence.

### 5 REAL CASE STUDY

The objective of our case study is to demonstrate the effect of missing values in semiconductor data. Without consideration of missing values, traditional tool commonality renders higher false identification rates. It will require engineers to spend extra time checking. We use a real case which consists of 73 wafers and 50 operations including several tools per operation to demonstrate the accuracy and efficiency of MCBMVE algorithm. On average, each operation contains five tools, with a maximum of 8 and minimum of 1. The total tool count is 57. In this case, we analyze the 22 suspected tools that were selected by engineers' domain knowledge.

#### 5.1 MCBMVE Demonstration on Root Cause

In this case, the root cause is Tool B1 of operation B, which is a single tool. In tool B1, it exists missing values as shown in Fig.3. We can see that the missing values include tool wafers passing through and the time period in which wafers passed through. However, the operation B only contains one single tool B1, so we can identify the missing value of the tool easily.

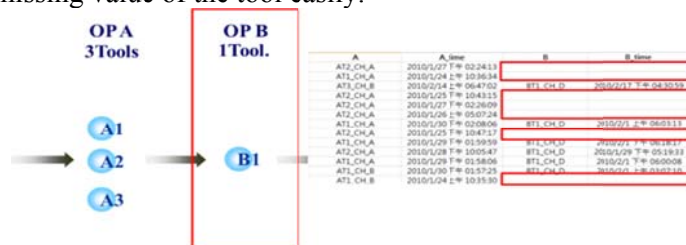


Figure 3: Missing Value in Tool B1

Without missing value consideration, the result of association rules is shown on the left side of Fig.4. The x axis represents conditional support, and the y axis represents the confidence, and each point represents the conditional support- confidence pair of tool. The red line is Peroto frontier; each point at the line is the non-dominated solution which has the highest ranking level. Furthermore, the conditional support-confidence pair will gain higher rank by higher support values. In addition, we rank other points by the distance to the Peroto frontier. The green color is the root cause (B1); the conditional support & confidence values are directly calculated as following:



$$S_{B1}^{direct} = \frac{N_{Bij}}{N_B} = \frac{13}{23} = 0.57 \tag{24}$$

$$C_{B1}^{direct} = \frac{N_{Bijk}}{N_{ijk}} = \frac{13}{13} = 1 \tag{25}$$

we can see that B1 will be screen out because the conditional support value is less than the minimal conditional support. In addition, there are 6 tools' conditional supports higher than the tool B1 such as tool A2. In this case, the blue circle highlights the most suspected tool (A2). The conditional support & confidence values are calculated as following:

$$S_{A2}^{direct} = \frac{N_{Bij}}{N_B} = \frac{21}{23} = 0.91 \tag{26}$$

$$C_{A2}^{direct} = \frac{N_{Bijk}}{N_{ijk}} = \frac{21}{21} = 1 \tag{27}$$

We can see that only 21 bad wafers passing through tool A2, but considering false identification in wafer pattern recognition, it is not the real root cause.

On the contrary, we go through the MCBMVE algorithm to complete the missing value of tool B1 100 times, and then re-calculate conditional support and confidence as following:

$$S_{B1}^{MCBMVE} = \frac{N_{Bij} + N_{Bij}^{(M)}}{N_B} = \frac{13+10}{23} = 1 \tag{28}$$

$$C_{B1}^{MCBMVE} = \frac{N_{Bijk} + N_{Bijk}^{(M)}}{N_{ijk} + N_{ijk}^{(M)}} = \frac{13+10}{13+10} = 1 \tag{29}$$

The conditional support-confidence pair is shown on the right side of Fig.4. We found that the conditional support-confidence pair of tool B1 is at Peroto frontier and tool B1 has the highest conditional support (1) and confidence value (1). It leads the ranking of tool B1 to rank 1.

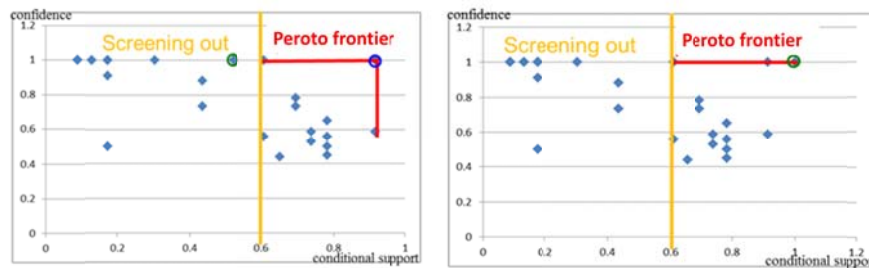


Figure 4: Result of Pareto Frontier

In this case, we compare the results with the traditional methods: RAR method and MVC method as shown in Table 3. We can see the results of the RAR method, both the conditional support and confidence values are revised to 1. The same phenomenon occurred in the MVC method, when both the conditional support and confidence values are revised to 1. These phenomenon show that our SBMVC method has the same performance as the two traditional methods in a single tool.

Table 3: Result of RAR, MVC Method of Op. B

Method	Result
<b>RAR</b>	$S_{B1}^{RAR} = \frac{N_{Bij}}{N_B} = \frac{N_{Bij}^{(O)}}{N_B - N_{Bi}^{(M)}} = \frac{13}{23 - 10} = 1$
	$C_{B1}^{RAR} = \frac{N_{Bijk}}{N_{ijk}} = \frac{N_{Bijk}^{(O)}}{N_{ijk}^{(O)}} = \frac{13}{13} = 1$
<b>MVC</b>	$S_{B1}^{MVC} = \frac{N_{Bij}}{N_B} = \frac{13 + 10}{23} = 1$
	$C_{B1}^{MVC} = \frac{N_{Bijk}}{N_{ijk}} = \frac{13 + 10}{13 + 10} = 1$

### 5.2 MCBMVE Demonstration on non-Root Cause

Besides operation B, we use another real case in semiconductor fabs to compare the accuracy and efficiency with RAR and MVC algorithm. We consider the data set in operation H, which consists of two tools and a total of 73 wafers. In operation H, there are 23 bad wafers, of which 12 are observed and 13 bad wafers are missing. The missing values include tool wafers passing through and the time period in which wafers passed through. Because the operation H is not a single tool, we cannot identify missing values of the tool. Without considering missing values, the conditional support & confidence values of tool H1 and tool H2 are calculated by a direct method as following:

$$S_{H1}^{direct} = \frac{N_{Bij}}{N_B} = \frac{5}{23} = 0.21 \tag{30}$$

$$C_{H1}^{direct} = \frac{N_{Bijk}}{N_{ijk}} = \frac{5}{13} = 0.38 \tag{31}$$

$$S_{H2}^{direct} = \frac{N_{Bij}}{N_B} = \frac{7}{23} = 0.3 \tag{32}$$

$$S_{H2}^{direct} = \frac{N_{Bijk}}{N_{ijk}} = \frac{7}{15} = 0.46 \tag{33}$$

We can see that tool H1 and tool H2 are less than the minimal conditional support, so they will be screened out.

In addition, the re-calculated conditional support & confidence values of tool H1 and tool H2 by MCBMVE method, and the expected value of conditional support & confidence are calculated as following:

$$S_{H1}^{MCBMVE,ave} = \frac{N_{Bij}}{N_B} = 0.41 \tag{34}$$

$$C_{H2}^{MCBMVE,ave} = \frac{N_{Bijk}}{N_{ijk}} = \frac{5}{13} = 0.43 \tag{35}$$

$$S_{H2}^{MCBMVE,ave} = \frac{N_{Bij}}{N_B} = 0.49 \tag{36}$$

$$C_{H2}^{MCBMVE,ave} = \frac{N_{Bijk}}{N_{ijk}} = \frac{5}{13} = 0.39 \tag{37}$$

We can found that the tool H1 and tool H2 will be screened out by the minimal conditional support.

After using MCBMVE algorithm to complete missing values, the tool that is not the root cause may not have largely changed in conditional support and confidence.

We compare the results of our method with two existing methods to deal with missing values, the RAR approach and the MVC approach. The result of tool H1 & H2 are shown in Table 4. We can see that the RAR method can obtain the new conditional support values, but cannot adjust the confidence value. Besides, the conditional support calculation via RAR become higher than that without missing value consideration. It is not desirable because the conditional supports are higher than the conditional support threshold significantly. Also, in semiconductor manufacturing, there are many cases for yield diagnosis with few wafers.

The same phenomenon is found in the MVC method. We can see the MVC method can obtain the new conditional support values, but the conditional support calculation via MVC become higher than that without missing value consideration. Besides, the conditional support and confidence value highly vary depending on the minimal confidence threshold value. Many tools often got the lower confidence without missing value consideration. MVC is not suitable for dealing with missing values in semiconductor because all the values are missing such as tool information and tool-time information. There are not enough rules to tell us how to obtain above information by MVC. In root cause diagnosis, both RAR method & MVC methods present high risk of false identification. For example, neither tool H1 nor tool H2 is the root cause, but the conditional support and confidence values are increased highly through RAR & MVC methods. It will cause a higher rank than the real root cause.

Table 4: Result of RAR, MVC Method of Op. H

Method	Result	
RAR	$S_{H1}^{RAR} = \frac{N_{Bij}}{N_B} = \frac{N_{Bij}^{(O)}}{N_B - N_{Bi}^{(M)}} = \frac{5}{23-11} = 0.42$	$S_{H2}^{RAR} = \frac{N_{Bij}}{N_B} = \frac{N_{Bij}^{(O)}}{N_B - N_{Bi}^{(M)}} = \frac{7}{23-11} = 0.64$
	$C_{H1}^{RAR} = \frac{N_{Bijk}}{N_{ijk}} = \frac{N_{Bijk}^{(O)}}{N_{ijk}^{(O)}} = \frac{5}{13} = 0.38$	$C_{H2}^{RAR} = \frac{N_{Bijk}}{N_{ijk}} = \frac{N_{Bijk}^{(O)}}{N_{ijk}^{(O)}} = \frac{7}{15} = 0.46$
MVC	$S_{H1}^{MVC} = \frac{N_{Bij}}{N_B} = \frac{5+11}{23} = 0.7$ <p><i>confidence &gt; threshold, <math>N_{Bij}^{(M,MVC)} = 11</math></i></p>	$S_{H2}^{MVC} = \frac{N_{Bij}}{N_B} = \frac{7+11}{23} = 0.78$ <p><i>confidence &gt; threshold, <math>N_{Bij}^{(M,MVC)} = 11</math></i></p>
	$C_{H1}^{MVC} = \frac{N_{Bijk}}{N_{ijk}} = \frac{5+11}{13+15} = 0.57$ <p><i>confidence &gt; threshold, <math>N_{ijk}^{(M,MVC)} = 15</math></i></p>	$C_{H2}^{MVC} = \frac{N_{Bijk}}{N_{ijk}} = \frac{7+11}{15+15} = 0.6$ <p><i>confidence &gt; threshold, <math>N_{ijk}^{(M,MVC)} = 15</math></i></p>

## 6 CONCLUSION

In industry practice, the association rule-based TCA is believed to be an effective approach to identifying tool excursions for yield enhancement. However, we found that missing values lead to high rates of false positive and false negative in association rule-based TCA. Thus, the recovery of missing values is an important issue to address before conducting TCA. The MCBMVE method proposed applies the Bernoulli distribution to derive the transition probabilities for updating the state probabilities of missing values until the convergence criterion is met. Comparing MCBMVE with the RAR or MVC method, this study found that the MCBMVE method has the same performance as RAR and MVC methods in root cause tools, and has better performance in non-root cause tools. The real case study shows that MCBMVE is more accurate in recovering missing values so as to improve the identification accuracy in TCA.

## ACKNOWLEDGMENTS

This work was supported in part by the National Science Council under NSC 98-2221-E002-138-MY3

## REFERENCES

- R. C. Leachman and S. Ding, S, Excursion yield loss and cycle time reduction in semiconductor manufacturing. *IEEE Transactions on Automation Science and Engineering*, 2011, vol. 8, 112-117.
- E.R.S. Pierre, E. Tuv, A. Borisov, Spatial Patterns in Sort Wafer Maps and Identifying Fab Tool Commonalities, *ASMC IEEE*, 2008, pp 268-272
- A. Ragel, B. Cre'milleux, Treatment of missing values for association rules, *Proceedings of The Second Pacific-Asia Conference on Knowledge Discovery and Data Mining*, Springer, Berlin, 1998 pp. 258–270.
- A.T. McCray, J. McNames and D. Abercrombie, Locating Disturbances in Semiconductor Manufacturing With Stepwise Regression, *IEEE Transactions on Semiconductor Manufacturing*, Aug. 2005, Vol. 18, p458 – 468
- R. H. Chen and C.M. Fan, Treatment of Missing Value for Association Rule-Based Tool Commonality Analysis in Semiconductor Manufacturing, *IEEE CASE 2012*, Aug. 2012

## AUTHOR BIOGRAPHIES

**CHIH-MIN FAN** is an Assistant Professor in the Department of Industrial Engineering and Management at Yuan Ze University, Taiwan. He received a master's degree in Electrical Engineering and a Ph.D. in the same subject from National Taiwan University, Taiwan. His current research and teaching interests are in statistical data analysis, knowledge representation and reasoning, engineering knowledge mining, and on-demand knowledge service with applications to engineering chain collaboration in semiconductor manufacturing industry.

**RONG-HUEI CHEN** received the Master degree in the Department of Electrical Engineering at National Taiwan University, Taiwan. He is currently a Ph.D student in the Electrical Engineering at National Taiwan University, Taiwan. His research interest is on the statistical data mining and semiconductor yield analysis, especially in the case of combinational effect in yield-loss event.