ELSEVIER

Expert Systems with Applications 29 (2005) 807–815

Expert Systems with Applications

www.elsevier.com/locate/eswa

# A novel manufacturing defect detection method using association rule mining techniques

Wei-Chou Chen, Shian-Shyong Tseng, Ching-Yao Wang*

*Department of Computer and Information Science, National Chiao Tung University, Hsinchu 300, Taiwan, ROC*

**Abstract**

In recent years, manufacturing processes have become more and more complex, and meeting high-yield target expectations and quickly identifying *root-cause machinesets*, the most likely sources of defective products, also become essential issues. In this paper, we first define the *root-cause machineset identification problem* of analyzing correlations between combinations of machines and the defective products. We then propose the *Root-cause Machine Identifier* (RMI) method using the technique of association rule mining to solve the problem efficiently and effectively. The experimental results of real datasets show that the actual root-cause machinesets are almost ranked in the top 10 by the proposed RMI method.


*Keywords:* Association rule mining; Defect detection; Interestingness measurement; Manufacturing defect detection problem

## 1. Introduction

In recent years, manufacturing processes have become more and more complex, and meeting high-yield target expectations and quickly identifying *root-cause machinesets*, the major killer machine(s) that causes a low-yield situation in a regular manufacturing procedure, also become essential issues. Although process control and statistical analysis techniques can be applied to establish a solid base for well-tuned manufacturing processes, identification of root-cause machineset is still hard and costly due to the existence of multiple coefficients among variants, nonlinear interactions, and the intermittent nature of the problem. For example, CIM/MES/EDA systems in most semiconductor manufacturing companies help users analyze collected manufacturing data in order to discover the root-cause machineset when a low-yield situation occurs; however, too many indexes and diagrams generated by the statistical methods in CIM/MES/EDA systems, such as K-W test, covariance analysis, regression analysis, etc., are usually not easy for engineers to assimilate and judge. On the other hand, lots of time is required to solve the false-alarm issue.

In this paper, we attempt to apply the technique of association rule mining to provide an efficient and effective solution. The *root-cause machineset identification problem* of analyzing correlations between combinations of machines and the defective products is first defined. Then the *Root-cause Machine Identifier* (RMI) method consisting of three phases, *data preprocessing*, *candidate generation* and *interestingness measurement*, is proposed to solve the problem. In the data preprocessing phase, two data preprocessing procedures, *stage-oriented* and *machine-oriented*, are proposed and can be selected at different considerations of manufacturing defect hypotheses to transform the raw data into the materials for mining. In the candidate generation phase, a level-wise processing procedure based on the Apriori property (Agrawl & Srikant, 1994) is used to remove the machinesets whose *defect converges* are less than the user-specified minimum defect coverage (i.e. they have not enough evidences to be the root cause) and generate the candidate machinesets for the transformed materials. The defect coverage of a machineset is defined as the percentage of all defective products passing through the target machineset. In the interestingness measurement phase, a user-specified interestingness measurement is used to evaluate the possibility of being the root cause for each candidate machineset. In additional to two typical interestingness measurements (*confidence* and

* Corresponding author. Tel.: +886 3571212 1 56658; fax: +886 3572 1490.

*E-mail addresses:* sirius@cis.nctu.edu.tw (W.-C. Chen), sstseng@cis.nctu.edu.tw (S.-S. Tseng), cywang@cis.nctu.edu.tw (C.-Y. Wang).

0957-4174/$ - see front matter © 2005 Elsevier Ltd. All rights reserved.
doi:10.1016/j.eswa.2005.06.004

$\phi$), an novel interestingness measurement considering the characteristic of continuity of defect products, called *continuity-based interestingness measurement*, is proposed and can be selected. Consequently, the candidate machinesets with their interestingness values are ranked in descending order and then provided to experts for further determination.

## 2. Related work

### 2.1. Data mining and mining association rules

Data mining, also referred to as 'knowledge discovery', means the process of extracting nontrivial, implicit, previously unknown and potentially useful information from databases (Chen, Han, & Yu, 1996; Han & Kamber, 2001). Depending on the types of knowledge derived, mining approaches may be classified as finding association rules (Agrawal, Imielinski, & Swami, 1993; Agrawl & Srikant, 1994; Brin, Motwani, & Silverstein, 1997; Brin, Motwani, Ullman, & Tsur, 1997; Cheung, Han, Ng, & Wong, 1996; Park, Chen, & Yu, 1995a; Park, Chen, & Yu, 1995b; Park, Chen, & Yu, 1995c; Wur & Leu, 1999), classification rules (Cheeseman & Stutz, 1996; Quinlan, 1986, 1993; Weiss & Kulikowski, 1991), clustering rules (Ester, Kriegel, & Xu, Kaufuman & Rousseeuw, 1990; Ng & Han, 1994; Zhang, Ramakrishnan, & Livny, 1996) and others (Catledge & Pitkow, 1995; Faloutsos, Ranganathan, & Manolopoulos, 1994; Han & Kamber, 2001). The most commonly seen is finding association rules in transaction databases.

Conceptually, an association rule indicates that the occurrence of certain items in a transaction would imply the occurrence of other items in the same transaction (Agrawal et al., 1993). The processing procedure for mining association rules can typically be decomposed into two tasks (Agrawl & Srikant, 1994): (a) discover the itemsets satisfying the user-specified minimum support from a given dataset, i.e. *finding frequent itemsets*, and (b) generate strong rules satisfying the user-specified minimum confidence from all frequent itemsets found by (a), i.e. *generating association rules*. Task (a) is used to obtain statistically significant patterns, and Task (b) is used to obtain interesting rules.

Since Task (a) is very time consuming compared to Task (b), the major challenge in mining association rules focuses on reducing the search space and decreasing the computation time in Task (a). Some famous mining algorithms were proposed to achieve this purpose. *Apriori* algorithm (Agrawl & Srikant, 1994), the best known, utilizes a level-wise candidate generation approach to reduce the search space such that only the frequent itemsets found in the previous level are used as seeds in generating the candidate itemsets in the current level. The key idea of the Apriori algorithm is that if an itemset does not satisfy the user-specified minimum support, then its proper supersets also will not and can be pruned. This Apriori property will greatly reduce the number of itemsets considered.

### 2.2. Interestingness measurement for association rules

Although a level-wise candidate generation algorithm can efficiently discover significant patterns, many of them may be not interesting to users. Thus, designing a useful interestingness measurement is becoming an important issue (Brin, Motwani et al., 1997; Chen et al., 1996; Han & Kamber, 2001; Tan & Kumar, 2000). *Confidence*, the most typical interestingness measurement for association rule mining, measures the conditional probability of events associated with a particular rule. For example, an association rule $A \rightarrow B$ with confidence $c\%$ means that $c\%$ of all transactions containing $A$ also contain $B$. However, the confidence measurement may be misleading or insufficient for many real-world applications. For example, given a minimum confidence of 60%, the association rule *milk* $\rightarrow$ *cigarette* with confidence 66% is then discovered in a supermarket. However, it is misleading since the probability of purchasing cigarette is 70%, which is even larger than 66%. In fact, milk and cigarette associate negatively since purchasing milk actually decreases the desirability of purchasing cigarettes. Thus, many researches (Brin, Motwani et al., 1997; Brin, Motwani, Ullman et al., 1997; Freitas, 1999; Hilderman & Hamilton, 1999; Piatestsky-Shaprio, 1991; Silberschatz & Tuzhilin, 1996; Tan & Kumar, 2000) have proposed other effective interestingness measurements.

Piatestsky-Shaprio (1991) proposed a domain-independent interestingness measurement to evaluate the interestingness of discovered rule $A \rightarrow B$:

$$\phi = \frac{|A\&B| - |A||B|/N}{\sqrt{|A||B|(1 - |A|/N)(1 - |B|/N)}},$$

where $N$ denotes the total number of tuples in the database, $|A|$ denotes the number of tuples that contain the antecedent $A$, $|B|$ denotes the number of tuples that contain the consequent $B$ and $|A\&B|$ denotes the number of tuples that contain both $A$ and $B$. The range of this interestingness measurement is between $-0.25$ and 0.25.

## 3. Root-cause machineset identification problem

Fig. 1 shows a general manufacturing process requiring a multistage production procedure. Each stage may have more than one machine performing the same task. Thus, products may pass through different machines in a specific stage.

Assume a shipment consists of $k$ identical products $\{p_1, p_2,\ldots,p_k\}$. Each product must pass through $l$ stages $\langle s_1, s_2,\ldots,s_l\rangle$ in sequence to be finished, and there are $n$ manufacturing machines $\{M_1, M_2,\ldots,M_n\}$ in this shipment. Note that a

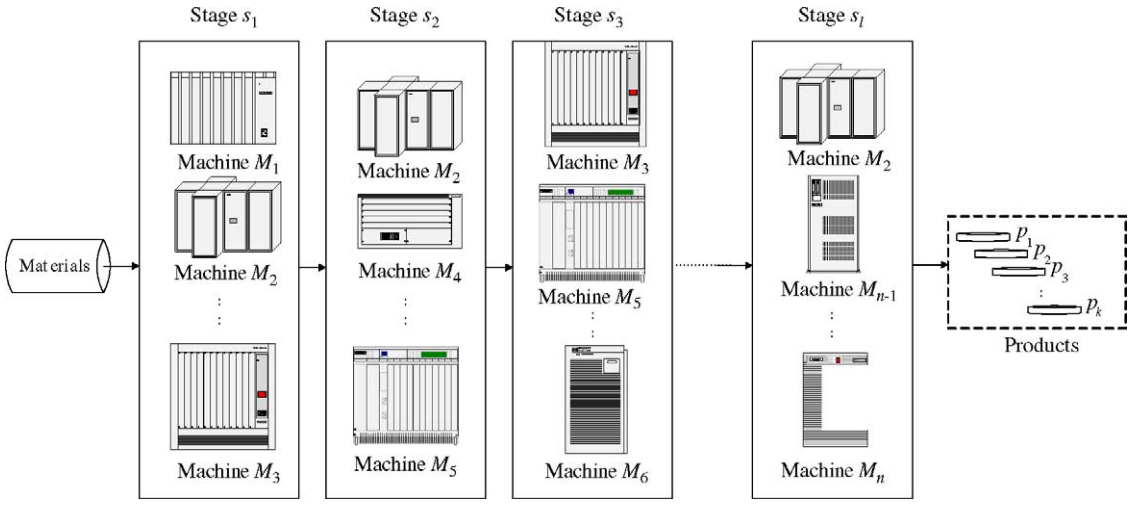


Fig. 1. A general manufacturing process.

machine with multiple functions may appear in more than one stage in the process. The *manufacturing process relation*, $r=\{t_1, t_2,\ldots,t_k\}$, based on the relation schema $R$(PID, $S_1$, $S_2,\ldots,S_l$, $D$), can be used to record the processing information from each stage and the test result for each product, $p_i$, $1\leq i\leq k$. Among the attributes in $R$, PID is an identification attribute used to uniquely label the products, $S_i$ is a context attribute associated with a pair ⟨*manufacturing machine, timestamp*⟩ used to indicate that the *manufacturing machine* is used in the *i*th stage at the *timestamp* for each product, and $D$ is a class attribute used to state whether a product is defective or not.

**Example 1**. Table 1 shows a manufacturing process relation used to record five-stage ($l=5$) and seven-machine ($n=7$) processing information for a shipment consisting of five products ($k=5$). The first tuple shows that product $p_1$ passed through stage 1 on ⟨$M_1$, 1⟩, stage 2 on ⟨$M_5$, 3⟩, stage 3 on ⟨$M_3$, 10⟩, stage 4 on ⟨$M_4$, 12⟩, and stage 5 on ⟨$M_5$, 14⟩, and its test result shows a defect ($D=$ 1). The other tuples have similar meanings.

Our goal is to identify the *root-cause machineset* for a given manufacturing process relation. In recent years, many approaches have been proposed to solve similar problems. Examples are such as Raghavan (2002) applied decision tree to discover the root cause of yield loss in integrated circuits, Gardner and Bieker (2000) combined self-organizing neural networks and rule induction to identify the critical poor yield factors from normally collected wafer manufacturing data, Mieno et al. (1999) applied a regression tree analysis to failure analysis in LSI manufacturing.

## 4. Root-cause machine identifier method

We attempt to apply the technique of association rule mining to solve the root-cause machineset identification problem. According to the general operation of mining association rules, three major scenarios need to be discussed:

(1) *Data preprocessing scenario*: Since the technique of association rule mining is usually performed on transactional data (its target of mining is not predetermined), it is important to transform the data in the manufacturing process relation into the materials and retain the *appropriate* relationships between machines and products that facilitate mining.
(2) *Mining procedure scenario*: A product may pass through hundreds of stages (machines) to be finished. The evaluation of all combinations of machines is relatively enormous and impractical. Therefore, the pruning strategy is required to remove the candidates with inadequate evidences to be the root cause such that the search space and the computation time can be reduced.

Table 1
A manufacturing process relation for five products in a five-stage manufacturing procedure

| PID | $S_1$ | $S_2$ | $S_3$ | $S_4$ | $S_5$ | $D$ |
|---|---|---|---|---|---|---|
| 1 | $M_1$, 1 | $M_5$, 3 | $M_3$, 10 | $M_4$, 12 | $M_5$, 14 | 1 |
| 2 | $M_2$, 5 | $M_1$, 8 | $M_1$, 12 | $M_2$, 15 | $M_1$, 17 | 0 |
| 3 | $M_3$, 2 | $M_3$, 7 | $M_5$, 13 | $M_4$, 17 | $M_3$, 20 | 0 |
| 4 | $M_3$, 4 | $M_1$, 6 | $M_4$, 14 | $M_4$, 18 | $M_5$, 19 | 1 |
| 5 | $M_4$, 7 | $M_2$, 11 | $M_4$, 15 | $M_2$, 20 | $M_5$, 23 | 1 |

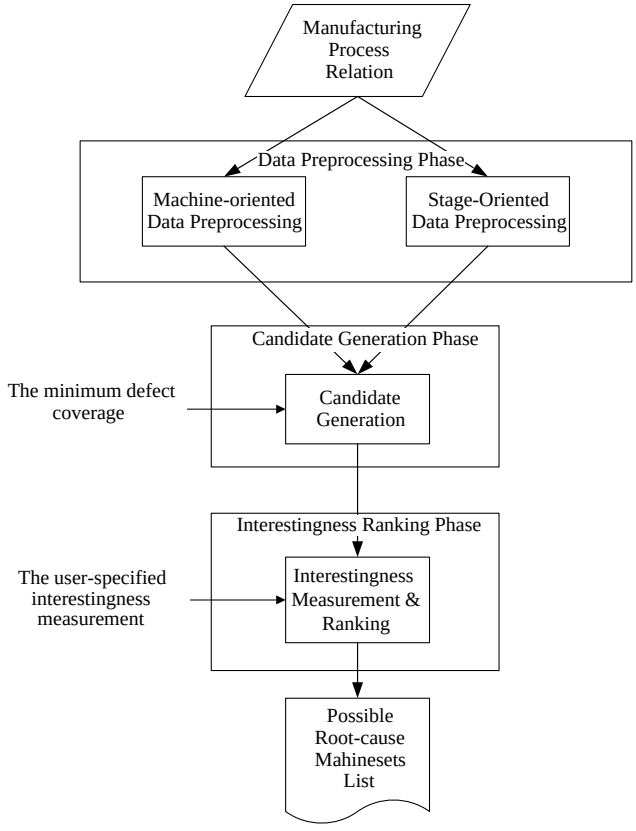


Fig. 2. The flowchart of the RMI method.

(3) *Visualization scenario*: Among the generated candidates, a suitable interestingness measurement is then needed to identify the root-cause machineset.

To overcome the above three scenarios, the RMI method shown in Fig. 2 consisting of three phases, *data preprocessing phase*, *candidate generation phase* and *interestingness ranking phase*, is proposed. The data preprocessing phase focuses on transforming the raw data in a given manufacturing process relation into transactional data. The candidate generation phase focuses on generating candidate machinesets from the transactional data, and the interestingness measurement phase focuses on identifying the root-cause machineset from the obtained candidate machinesets.

By the user-selected preprocessing procedure in the data preprocessing phase, the RMI method first gets materials transformed from the data in the manufacturing process relation. Then given a user-specified *minimum defect coverage*, a threshold used to remove the machinesets without enough evidences to be the root cause, the RMI method generates all candidate machinesets by the candidate generation phase. Finally, by the interestingness ranking phase, the RMI method ranks the candidate machinesets based upon a user-specified interestingness measurement and provides the result to experts for further determination.

Table 2
An example of the machine-oriented preprocessing procedure

| TID | Items |
|---|---|
| 1 | $M_1$, $M_3$, $M_4$, $M_5$ |
| 4 | $M_1$, $M_3$, $M_4$, $M_5$ |
| 5 | $M_2$, $M_4$, $M_5$ |

### 4.1. Data preprocessing phase

The data preprocessing phase first selects the defective tuples from a given manufacturing process relation. Two data preprocessing procedures, *machine-oriented* and *stage-oriented* preprocessing procedures, have been proposed to handle different manufacturing defect hypotheses. The machine-oriented preprocessing procedure concentrates on the machines a product passes through, regardless of the manufacturing stage. Thus, although a machine may be used in more than one stage in a tuple because of its multi-functionality, this preprocessing procedure treats it as only a single appearance.

**Example 2**. For the manufacturing process relation shown in Table 1, the machine-oriented preprocessing procedure transforms the defect tuples 1, 4 and 5 as shown in Table 2. The tuple TID1 = $\{M_1, M_2, M_4, M_5\}$ means that the product $p_1$ passed through four machines, $M_1$, $M_3$, $M_4$ and $M_5$. The other tuples have similar meanings.

The machine-oriented preprocessing procedure transforms the processing information in the manufacturing process relation into intuitive transactional data and assumes a machine's functions are correlated. That is, if one function is faulty, the other may also be. By contrast, the stage-oriented preprocessing procedure assumes that a machine's functions are not correlated. If one function is faulty, the other ones may still operate normally. Therefore, this preprocessing procedure treats machines in different stages as distinct individuals.

**Example 3**. For the manufacturing process relation shown in Table 1, the stage-oriented preprocessing procedure transforms the defect tuples 1, 4 and 5 as shown in Table 3. The machine $m_{11}$ indicating $M_1$ is used at stage 1 is different from the machine $m_{12}$ indicating $M_1$ is used at stage 2. The tuple TID1 = $\{m_{11}, m_{52}, m_{33}, m_{44}, m_{55}\}$ means that the

Table 3
An example of the stage-oriented preprocessing procedure

| TID | Items |
|---|---|
| 1 | $m_{11}$, $m_{52}$, $m_{33}$, $m_{44}$, $m_{55}$ |
| 4 | $m_{31}$, $m_{12}$, $m_{43}$, $m_{44}$, $m_{55}$ |
| 5 | $m_{41}$, $m_{22}$, $m_{43}$, $m_{24}$, $m_{55}$ |

product $p_1$ passed through stage 1 on $M_1$, stage 2 on $M_5$, stage 3 on $M_3$, stage 4 on $M_4$ and stage 5 on $M_5$. The other tuples have similar meanings.

### 4.2. Candidate generation phase

A level-wise processing procedure like finding frequent itemsets in association rules mining is used to generate possible sets of machines called *candidate machinesets*. The *defect coverage* of a machineset is defined as the percentage of all defective products passing through the target machineset. Therefore given the user-specified *minimum defect coverage*, in the first iteration, the proposed candidate generation phase calculates the defect coverage for each individual machine, and then retains the 1-machinesets that satisfy the minimum defect coverage as candidates. In the second iteration, the proposed phase generates machinesets consisting of two machines by joining the candidate 1-machinesets from the first iteration, and retains the 2-machinesets that satisfy the minimum defect coverage as candidates. In each subsequent iteration, candidate machinesets found in the preceding iteration are used as seeds in the current iteration, and the process continues until no new candidate machinesets can be generated.

Since this level-wise processing procedure is based on the Apriori property, each proper subset of a candidate machineset must be a candidate. In other words, if a machineset does not satisfy the user-specified minimum defect coverage, then none of its proper supersets will be. This can greatly reduce the number of candidate machinesets to be considered. Moreover, to improve the computation performance, the candidate generation phase retains defective product information for each candidate machineset in the current level so that each machineset's defect coverage information in the next level can be efficiently calculated by utilizing the retained information rather than re-processing the original database.

**Example 4**. Table 4 shows the defect coverage for each 1-machineset in Table 3. The first tuple shows that only the defective product $p_1$ passed through the machineset $m_{11}$. Thus, the defect coverage of $m_{11}$ is 1/3 = 33%.

Table 4
Defect coverage and defective product information for each 1-machineset in Table 3

| Machineset | Involved defective products | Defect coverage (%) |
|---|---|---|
| $m_{11}$ | $p_1$ | 33 |
| $m_{31}$ | $p_4$ | 33 |
| $m_{41}$ | $p_5$ | 33 |
| $m_{52}$ | $p_1$ | 33 |
| $m_{12}$ | $p_4$ | 33 |
| $m_{22}$ | $p_5$ | 33 |
| $m_{33}$ | $p_1$ | 33 |
| $m_{43}$ | $p_4$, $p_5$ | 66 |
| $m_{44}$ | $p_1$, $p_4$ | 66 |
| $m_{24}$ | $p_5$ | 33 |
| $m_{55}$ | $p_1$, $p_4$, $p_5$ | 100 |

Table 5
Defect coverage and defective product information for each candidate 1-machineset obtained

| Machineset | Involved defective products | Defect coverage (%) |
|---|---|---|
| $m_{43}$ | $P_4$, $p_5$ | 66 |
| $m_{44}$ | $P_1$, $p_4$ | 66 |
| $m_{55}$ | $P_1$, $p_4$, $p_5$ | 100 |

Table 6
Defect coverage and defective product information for each 2-machinesets generated

| Machineset | Involved defective products | Defect coverage (%) |
|---|---|---|
| $m_{43}$, $m_{44}$ | $p_4$ | 33 |
| $m_{43}$, $m_{55}$ | $p_4$, $p_5$ | 66 |
| $m_{44}$, $m_{55}$ | $p_1$, $p_4$ | 66 |

**Example 5**. Continuing from Example 4 and assuming the user-specified minimum defect coverage is 50%, Table 5 shows candidate 1-machinesets of Table 4.

Next, 2-machinesets $\{m_{43}, m_{44}\}$, $\{m_{43}, m_{55}\}$ and $\{m_{44}, m_{55}\}$ are then generated by joining the candidate 1-machinesets in Table 5. The defect coverage for $\{m_{43}, m_{44}\}$ is 33% and its defective product information is $\{p_4\}$ by performing the intersection of the set of defective products of $m_{43}$ and $m_{44}$. Complete results are shown in Table 6.

As we can see, the machineset $\{m_{43}, m_{44}\}$ is removed since its defect coverage is less than 50%, the specified minimum defect coverage. The resulting candidate 2-machinesets are shown in Table 7.

Next, the only 3-machineset $\{m_{43}, m_{44}, m_{55}\}$ generated by joining the candidate 2-machinesets in Table 7.

Table 7
Defect coverage and defective product information for each candidate 2-machineset obtained

| Machineset | Involved defective products | Defect coverage (%) |
|---|---|---|
| $m_{43}$, $m_{55}$ | $p_4$, $p_5$ | 66 |
| $m_{44}$, $m_{55}$ | $p_1$, $p_4$ | 66 |

Table 8
Defect coverage and defective product information for each candidate machinesets obtained

| Machineset | Involved defective products | Defect coverage (%) |
|---|---|---|
| $m_{43}$ | $p_4$, $p_5$ | 66 |
| $m_{44}$ | $p_1$, $p_4$ | 66 |
| $m_{55}$ | $p_1$, $p_4$, $p_5$ | 100 |
| $m_{43}$, $m_{55}$ | $p_4$, $p_5$ | 66 |
| $m_{44}$, $m_{55}$ | $p_1$, $p_4$ | 66 |

However, since $\{m_{43}, m_{44}\}$ is not included in the set of candidate 2-machinesets, it is removed according to above-mentioned Apriori property. All candidate machinesets generated are shown in Table 8.

### 4.3. Interestingness ranking phase

Although a candidate machineset having high defect coverage is statistically significant, it may not have a high possibility of being the root cause. For example, the defect coverage of $m_{43}$ is the same as that of $m_{44}$ in Table 8, but intuitively, $m_{43}$ is more probable than $m_{44}$ since all products passing through it are defective. In this section, an interestingness ranking phase using an interestingness measurement to evaluate correlations between candidate machinesets and defective products is proposed for finding the root-cause machineset. Below, in additional to two typical interestingness measurements *confidence* and $\phi$, an novel interestingness measurement called *continuity-based interestingness measurement* is proposed to extend $\phi$.

*Confidence*, the most well-known interestingness measurement for association rule mining, calculates the conditional probability that a candidate machineset causes defective products (*machineset*→*defect*). That is, it calculates the percentage of all products passing through a candidate machineset that are defective. $\phi$, a domain-independent interestingness measurement proposed by Piatestsky-Shaprio (1991) evaluates the discovered rule $A \rightarrow B$ as follows:

$$\phi = \frac{|A\&B| - |A||B|/N}{\sqrt{|A||B|(1-|A|/N)(1-|B|/N)}}.$$

This equation indicates the degree to which 'when antecedent $A$ appears, consequent $B$ also appears'. If $A$ is regarded as a certain candidate machineset and $B$ is regarded as a defective product, then the equation calculates the degree of correlation between the candidate machineset and the defect.

However, the manufacturing process characteristics, such as the observation that the root-cause machineset often produces defective products continuously, are not considered in the two above-mentioned interestingness measurements. Thus, we propose *continuity* function to measure the continuity between the defective products for a candidate machineset. High continuity may indicate a higher probability of being the root cause. We can easily extend the interestingness measurement $\phi$ to $\phi'$, called *continuity-based interestingness measurement*, as follows:

$$\phi' = \phi * \text{ continuity}.$$

The *continuity* function calculates the reciprocal of the average distance between pairs of neighboring defective products in the product sequence as follows:

$$\begin{cases} \text{Continuity} = 0 & \text{if } |X| \le 1 \\ \text{Continuity} = \dfrac{1}{\sum_{i=1}^{i=|X|-1} d(\alpha(x_i), \alpha(x_{i+1}))/|X| - 1} & \text{if } |X| > 1, \end{cases}$$

where $X=(x_1, x_2,\ldots)$ denotes a defective product sequence contained in the product sequence $P=(p_1, p_2,\ldots)$ which is a sequence of products passing through a candidate machineset (i.e. $X$ is a subsequence of $P$), $|X|$ denotes the number of defective products, $\alpha(x_i)$ denotes the order of the defective product $x_i$ in $P$ (e.g. if $\alpha(x_i)=j$, $x_i$ is the $j$th product in $P$), and $d(\alpha(x_i),\alpha(x_{i+1}))$ is the distance of $\alpha(x_i)$ and $\alpha(x_{i+1})$, which can easily be calculated by $\alpha(x_{i+1})-\alpha(x_i)$.

Table 9
Calculated continuities for each candidate machineset in Table 8

| Machineset | Product sequence | Defective product sequence | Continuity |
|---|---|---|---|
| $m_{43}$ | $(p_4, p_5)$ | $(p_4, p_5)$ | 1 |
| $m_{44}$ | $(p_1, p_3, p_4)$ | $(p_1, p_4)$ | 0.5 |
| $m_{55}$ | $(p_1, p_4, p_5)$ | $(p_1, p_4, p_5)$ | 1 |
| $m_{43}, m_{55}$ | $(p_4, p_5)$ | $(p_4, p_5)$ | 1 |
| $m_{44}, m_{55}$ | $(p_1, p_4)$ | $(p_1, p_4)$ | 1 |

**Example 6**. Table 9 shows the product sequence, defective product sequence, and calculated continuity value for each candidate machineset in Table 8. Among them, the continuity value of $m_{44}$ is $1/(d(\alpha(p_1), \alpha(p_4))/(2-1) = 0.5$ according to its product sequence $(p_1, p_3, p_4)$ and defective product sequence $(p_1, p_4)$.

According to the user-specified interestingness measurement, the set of candidate machinesets with their interestingness values are ranked in descending order.

**Example 7**. Continuing from Example 6, Table 10 shows the $\phi'$ for each candidate machineset. Since $m_{55}$ has highest interestingness value, the machine $M_5$ is the most likely the root-cause machineset.

Table 10
$\phi'$ for each candidate machinesets in Table 8

| Machineset | $\phi$ | Continuity | $\phi'$ |
|---|---|---|---|
| $m_{43}$ | 0.67 | 1 | 0.67 |
| $m_{44}$ | 0.167 | 0.5 | 0.0835 |
| $m_{55}$ | 1 | 1 | 1 |
| $m_{43}, m_{55}$ | 0.67 | 1 | 0.67 |
| $m_{44}, m_{55}$ | 0.67 | 1 | 0.67 |

Table 11
Relevant information for the nine real datasets

| Dataset | Data size (Products*stages) | Number of machines in machine-oriented preprocessing procedure | Number of machines in stage-oriented preprocessing procedure |
|---|---|---|---|
| Case 1 | 153*658 | 368 | 2727 |
| Case 2 | 145*867 | 497 | 4509 |
| Case 3 | 141*837 | 499 | 4434 |
| Case 4 | 116*624 | 416 | 2500 |
| Case 5 | 305*733 | 424 | 3094 |
| Case 6 | 53*587 | 411 | 2414 |
| Case 7 | 484*709 | 455 | 3381 |
| Case 8 | 106*632 | 419 | 2618 |
| Case 9 | 77*1109 | 450 | 3367 |

Table 12
Accuracy results of the RMI method for the nine datasets

| Dataset | Machine-oriented preprocessing procedure | | | Stage-oriented preprocessing procedure | | |
|---|---|---|---|---|---|---|
| | Minimum defect coverage = 0.3 Rank | Minimum defect coverage = 0.4 Rank | Minimum defect coverage = 0.5 Rank | Minimum defect coverage = 0.3 Rank | Minimum defect coverage = 0.4 Rank | Minimum defect coverage = 0.5 Rank |
| Case 1 | 4 | 4 | 4 | 22 | 12 | 6 |
| Case 2 | 1 | 1 | 1 | 1 | 1 | 1 |
| Case 3 | 1 | 1 | 1 | 1 | 1 | 1 |
| Case 4 | 1 | 1 | 1 | 1 | 1 | 1 |
| Case 5 | 1 | 1 | 1 | 1 | 1 | 1 |
| Case 6 | 106 | 93 | 78 | 145 | 90 | 58 |
| Case 7 | 6 | 5 | 5 | 2 | 1 | 1 |
| Case 8 | 51 | 47 | 40 | 43 | 23 | X |
| Case 9 | 74 | 50 | 44 | 10 | X | X |

## 5. Experimental results

The RMI method was implemented in Java on a Pentium-IV 2.4 G processor desktop with 512 MB RAM, and nine real datasets with the known root-cause machineset provided by the *Taiwan Semiconductor Manufacturing Corporation* (TSMC) were used to evaluate its accuracy. As shown in Table 11, 368 and 2727 machines needed to be considered in machine-oriented and stage-oriented preprocessing procedures, respectively, for Case 1 having 153 products and each passing through 658 stages.

With the minimum defect coverages ranging from 0.3 to 0.5 and the interestingness measurement $\phi'$, the ranks of the actual root-cause machinesets among the generated candidate machinesets are shown in Table 12. For example, the rank of the actual root-cause machineset for Case 1 was the 4th using machine-oriented preprocessing procedure with the minimum defect coverage = 0.3. Note that 'X' means the actual root-cause machineset cannot be found by the proposed method.

As stated previously, the machine-oriented preprocessing procedure assumes all functions of a machine are co-affected whereas the stage-oriented preprocessing procedure assumes each function of a machine is independent. Table 12 shows that the RMI method seems to have higher accuracy with the stage-oriented preprocessing procedure than with the machine-oriented preprocessing procedure in this semiconductor manufacturing experiment, if appropriate minimum defect coverages were set. By consulting with the product engineers for all above cases, the explanations of the experimental results are concluded as follows:

(a) For Cases 2, 3, 4 and 5, the actual root-cause machinesets were all ranked in the first place both with the machine-oriented and the stage-oriented preprocessing procedures. The major reasons are: (a) for Cases 2 or 3, the actual root-cause machineset was a single-function machine. Therefore, it had the same interestingness value both with the stage-oriented and machine-oriented preprocessing procedures; (b) for Cases 4 or 5, most functions of the actual root-cause machineset had high interestingness values and were ranked in the top 10 with the stage-oriented preprocessing procedure. Therefore, on the whole, the actual root-cause machineset with the machine-oriented preprocessing procedure still had a not-bad rank.
(b) For Cases 6, 7, 8, or 9, many normal products passed through the actual root-cause machineset without passing through the faulty function. Therefore the actual root-cause machineset had higher rank with the stage-oriented preprocessing procedure than with the machine-oriented preprocessing procedure, if an appropriate minimum defect coverage was set.

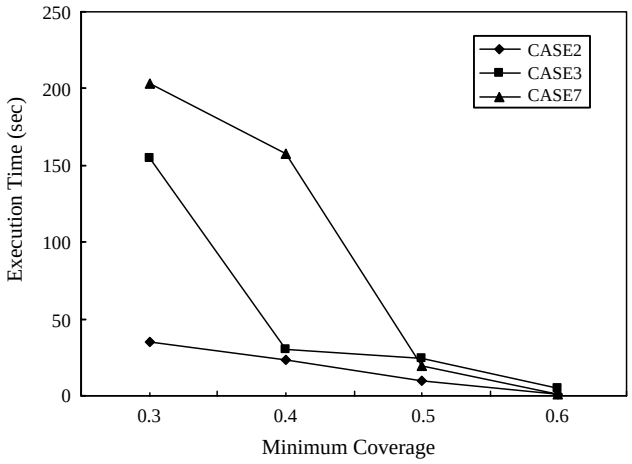


Fig. 3. Execution times for Case 2, Case 3 and Case 7 with the minimum defect coverage set from 0.3 to 0.6.

(c) For Case 1, the actual root-cause machineset had the same interestingness value in the machine-oriented and stage-oriented preprocessing procedures because it is a single-function machine (as in Cases 2 and 3). However, since most of the other candidate machinesets had lower interestingness values with the machine-oriented preprocessing procedure, the actual root-cause machineset with this preprocessing procedure had higher rank than with the stage-oriented preprocessing procedure. This was a special case in our experiments.

The actual root-cause machineset in most cases was ranked in the top 10 with an appropriate minimum defect coverage, except in Case 6, which had only 53 products so the actual root-cause machineset was not more significant than the others. Intuitively, setting a higher minimum defect coverage will prune more machinesets from consideration during the candidate generation phase, and thus decrease the execution time. As shown in Table 12 and Fig. 3, the higher minimum defect coverage is, the higher performance that RMI method can be. However, the RMI method may prune the actual root-cause machinesets out once the minimum defect coverage is set too high. How to set appropriate minimum defect coverage is thus becoming a critical issue for future investigation.

In order to demonstrate the accuracy of $\phi'$ compared to other known interestingness measures, Table 13 shows the rank of the actual root-cause machineset among all candidate machinesets generated by the RMI method when associated with three interestingness measures, *confidence*, $\phi$ and $\phi'$. The result shows that our proposed interestingness measurement $\phi'$ did not always outperform $\phi$ or *confidence* since the properties of all given testing cases were different, and that continuity can highlight cases 1, 7 and 9 with strong continuity defect signal.

## 6. Conclusion

Identification of the root-cause machineset in manufacturing can not only reduce manufacturing costs, but also improve manufactory performance. However, conventional methodologies for identifying root causes are restricted and dependent on experience and expertise. In this paper, we have defined the *root-cause machineset identification problem* and proposed RMI method to solve the problem efficiently and effectively. Two different data preparation procedures have proposed to transform the raw data into the desired format based on different manufacturing defect hypotheses. Also, an novel interestingness measurement considering the manufacturing continuity has proposed for the interestingness measurement phase in RMI method. Currently, the proposed RMI method has been considered as one of standard component in semiconductor manufacturing defect detection solution using data mining techniques of SAS® Taiwan Cooperation in order to help FAB users discover root causes. The experimental results show that about 80% cases can be ranked at the top ten and 20% cases are still remained unsolvable. In the future, we will continue our research to refine interestingness measurements of RMI method, and

Table 13
Accuracy results of the RMI method on the nine datasets for interestingness measurements *confidence*, $\phi$ and $\phi'$

| Dataset | Machine-oriented preprocessing procedure (minimum defect coverage = 0.3) | | | Stage-oriented preprocessing procedure (minimum defect coverage | | |
|---|---|---|---|---|---|---|
| | Confidence Rank | $\phi$ Rank | $\phi'$ Rank | Confidence Rank | $\phi$ Rank | $\phi'$ Rank |
| Case 1 | 8 | 4 | 4 | 41 | 17 | 22 |
| Case 2 | 1 | 1 | 1 | 1 | 1 | 1 |
| Case 3 | 1 | 1 | 1 | 1 | 1 | 1 |
| Case 4 | 1 | 1 | 1 | 1 | 1 | 1 |
| Case 5 | 1 | 1 | 1 | 3 | 1 | 1 |
| Case 6 | 163 | 94 | 106 | 168 | 128 | 145 |
| Case 7 | 9 | 8 | 6 | 1 | 4 | 2 |
| Case 8 | 25 | 32 | 51 | 2 | 2 | 43 |
| Case 9 | 114 | 57 | 74 | 46 | 22 | 10 |

develop automatic/semi-automatic mechanisms to solve the low-yield situations.

## Acknowledgements

This research was supported by the National Science Council of the Republic of China under Grant No. NSC93-2752-E-009-006-PAE.

## References

Agrawal, R., Imielinski, T., & Swami. A. (1993). Mining association rules between sets of items in large database. In: *ACM SIGMOD Conference*, May 1993, pp. 207–216.

Agrawl, R., & Srikant, R. (1994). Fast algorithm for mining association rules. In: *ACM VLDB Conference*, September 1994, pp. 487–499.

Brin, S., Motwani, R., & Silverstein, C. (1997). Beyond market basket: generalizing association rules to correlations. In: *ACM SIGMOD Conference*, Tucson, Arizona, USA, pp. 265–276.

Brin, S., Motwani, R., Ullman, J. D., & Tsur, S. (1997). Dynamic itemset counting and implication rules for market basket data. In: *ACM SIGMOD Conference*, Tucson, Arizona, USA, pp. 255–264.

Catledge, L. D., & Pitkow, J. E. (1995). Characterizing browsing strategies in the World Wide Web. In: *Proceedings of Third WWW Conference*, April 1995.

Cheeseman, P., & Stutz, J. (1996). Bayesian classification (AutoClass): theory and results. In U. M. Fayyad, G. Piatetsky-Shaprio, P. Smyth, & R. Uthurusamy (Eds.), *Advances in knowledge discovery and data mining* (pp. 153–180). AAAI/MIT Press, 153–180.

Chen, M. S., Han, J., & Yu., P. S. (1996). Data mining: an overview from a database perspective. *IEEE Transactions on Knowledge and Data Engineering*, *8*(6).

Cheung, D. W., Han, J., Ng, V. T., & Wong, C. Y. (1996). Maintenance of discovered association rules in large databases: an incremental updating approach. In: *IEEE International Conference on Data Engineering*, pp. 106–114.

Ester, M., Kriegel, H. O., & Xu, X. Knowledge discovery in large spatial databases: focusing techniques for efficient class identification. In: *Proceedings of Fourth International Symposium on Large Spatial Databases*, Portland, pp. 67–82.

Faloutsos, C., Ranganathan, M., & Manolopoulos, Y. (1994). Fast subsequence matching in time-series databases. In: *Proceedings of ACM SIGMOD Conference*, pp. 419–429.

Freitas, A. A. (1999). On rule interestingness measures. *Knowledge-Based System* , 309–315.

Gardner, M., & Bieker, J. (2000). Data mining solves tough semiconductor manufacturing problems. In: *ACM KDD Conference*, Boson, USA.

Han, J., & Kamber, M. (2001). *Data mining: concepts and techniques*. Los Altos, CA: Morgan Kaufmann.

Hilderman, R. J., & Hamilton, H. J. (1999). Heuristic measures of interestingness. *Principles of Data Mining and Knowledge Discovery* , 232–241.

Kaufuman, L., & Rousseeuw, P. J. (1990). *Finding groups in data: an introduction to cluster analysis*. New York: Wiley.

Mieno, F., Santo, T., Shibuya, Y., Odagiri, K., Tsuda, H., & Take, R. (1999). Yield improvement using data mining system. In: *IEEE Semiconductor Manufacturing Conference*.

Ng, R., & Han, J. (1994). Efficient and effective clustering method for spatial data mining. In: *ACM VLDB Conference*, Santiago, Chile, September 1994, pp. 144–155.

Park, J. S., Chen, M. S., & Yu, P. S. (1995a). An effective hash-based algorithm for mining association rules. In: *ACM SIGMOD Conference*, San Jose, CA, pp. 175–186.

Park, J. S., Chen, M. S., & Yu, P. S. (1995b). Mining association rules with adjustable accuracy. *IBM Research Report* .

Park, J. S., Chen, M. S., & Yu, P. S. (1995c). Efficient parallel data mining for association rules. In: *ACM CIKM Conference*, pp. 175–186.

Piatestsky-Shaprio, G. (1991). Discovery, analysis and presentation of strong rules. In G. Piatetsky-Shapiro, & W. J. Frawley (Eds.), *Knowledge discovery in databases* (pp. 229–247). AAAI, 229–247.

Quinlan, J. R. (1986). Induction of decision trees. *Machine Learning*, *1*, 81–106.

Quinlan, J. R. (1993). *C4.5: Programs for Machine Learning*. Los Altos, CA: Morgan Kaufmann.

Raghavan, V. (2002). Application of decision trees for integrated circuit yield improvement. In: *IEEE/SEMI Advanced Semiconductor Manufacturing Conference and Workshop*.

Silberschatz, A., & Tuzhilin, A. (1996). What makes patterns interesting in knowledge discovery systems. *IEEE Transaction on Knowledge and Data Engineering* .

Tan, P. N., & Kumar, V. (2000). Interestingness measures for association patterns: a perspective. In: *KDD'2000 Workshop on Postprocessing in Machine Learning and Data Mining*, Boston, MA.

Weiss, S. M., & Kulikowski, C. A. (1991). *Computer systems that learn: classification and prediction methods from statistics, neural nets, machine learning and expert systems*. Los Altos, CA: Morgan Kaufman.

Wur, S. Y., & Leu, Y. (1999). An effective Boolean algorithm for mining association rules in large databases. In: *International Conference on Database Systems for Advanced Applications (DASFAA '99)*, Hsinchu, Taiwan.

Zhang, T., Ramakrishnan, R., & Livny, M. (1996). BIRCH: an efficient data clustering method for very large databases. In: *ACM SIGMOD International Conference Management of Data*, Montreal, Canada, pp. 103–114.