Feature Selection Technique to Improve Performance Prediction in a Wafer Fabrication Process

KITTISAK KERDPRASOP and NITTAYA KERDPRASOP
Data Engineering Research Unit,
School of Computer Engineering, Suranaree University of Technology,
111 University Avenue, Nakhon Ratchasima 30000
THAILAND
kittisakThailand@gmail.com, nittaya@sut.ac.th

Abstract: - Fabrication of integrated circuits onto the raw silicon wafers is one major task of most complex semiconductor manufacturing. The fabrication process consists of a series of steps to cover special material layers over the wafer surface. Wafers re-enter the same processing machines as each layer is successively covered. Some defects in this complicated process can make the final products fail the test. Early fault detection during this critical manufacturing process can obviously improve product quality and reliability. However, timely yield analysis of large scale and multi-dimensional data that are constantly and automatically generated from hundreds of operational units in the production line of modern semiconductor industry is a challenging problem for process engineers. In this paper, we propose a novel feature selection technique to reduce dimensions of data and also to increase the prediction accuracy of the induced model.

Key-Words: - Feature selection technique, Manufacturing process, Intelligent manufacturing, Sequence analysis, Data mining, Semiconductor, Wafer fabrication process, Performance prediction.

1 Introduction

Semiconductor manufacturing is a highly complex production process composed of hundreds of steps. The major processes in most semiconductor industries are: production of silicon wafers from pure silicon material, fabrication of integrated circuits onto the raw silicon wafers, assembly by putting the integrated circuit inside a package to form a ready-to-use product, and testing of the finished products [1]. A constant advancement in the semiconductor industry is due mainly to persistent improvement of the wafer fabrication process.

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

Wafer fabrication in the semiconductor industry is probably one of the most complex manufacturing processes. Maintaining high yields through the statistical process control as a sole monitoring method for quality control is obviously inefficient in such highly complicated operations. Recent trend in intelligent manufacturing [2], [3] is to apply data mining techniques to automatically identify patterns and causal relationships leading to poor yield. Applying data mining technique to such high dimensional data is however not a straightforward task because the induced patterns are normally low accurate in their predictive performances. In this paper, we thus present a feature selection technique to remedy the high dimensional problem and also to help improving the discovery of patterns to detect tool fault that leads to low performance of a wafer lot from the semiconductor fabrication process.

2 Related Work

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

Ison and colleagues [4] proposed a decision tree classification model to detect fault of plasma etch
equipment. The model was built from the five sensor signal data. Goodlin et al [5] proposed to build a specific control chart for detecting specific type of faults. They collected tool-state data directly from the etcher. These data consist of 19 variables. The work of Spitzelsperger and colleagues [6] was also based on the statistical method. They adopted the multivariate control chart method to maintain changes in the mean and standard deviation coefficients by remodeling technique.

Later interest in fault detection has been shifted toward the non-parametric approaches. He and Wang [7] proposed to use the k-nearest neighbor rule for fault detection. Verdier and Ferreira [8] also applied the k-nearest neighbor method, but they proposed to use the adaptive Mahalanobis distance instead of the Euclidean distance. Tafazzoli and Saif [9] proposed a combined support vector machine methodology for process fault diagnosis. Ge and Song [10] applied support vector data to the principal component analysis method to detect process abnormalities.

Most work on fault detection methods has studied the process control problem with a few features of tool-state and process-state measurement data. McCann and his team [11] proposed a rather different setting in which the measurement data from the wafer fabrication process contain as much as 590 features. They applied feature selection technique to select only 40 features for further analysis.

In this work, we also apply a feature selection step prior to the application of data mining technique. Our feature selection is based on Bayesian analysis.

3 Feature Selection and Data Analysis Methodologies

3.1 Data Set

In our product performance analysis, we use the data set named SETFI (Semiconductor Tool level Fault Isolation), which is a simulated dataset [12] that closely mimics the actual high complexity of semiconductor manufacturing process. The data set contains 4000 records of the wafer fabrication process. During the process each wafer goes through sequence of operations in batch, which is called lot in this data set. The sequences of hundreds of operations might be different from lot to lot, but these operations involve only twenty tools, number 1 to 20. At each operation unit, only a single tool is in operation.

The original SETFI dataset contains the tool number applied in each of the 300 operational units together with the timestamps of each operation. In this study, we remove the first column (Lot #) because it plays no role to the discovering of performance patterns. We also ignore the timestamps because our main objective is the categorical analysis, not a time series analysis.

At the end of the fabrication process, a number of inspection steps are carried out to measure the product performance. Wafer lots that fail the inspection tests need re-processing. Low performance metric is often caused by a small subset of tools. Identifying such problematic tools at an early stage can obviously improve yield performance of the semiconductor manufacturing. Some data instances of the SETFI dataset are shown in Table 1.

Table 1. Example of five data instances in the SETFI data set.

<table>
<thead>
<tr>
<th>Lot #</th>
<th>Operation1</th>
<th>...</th>
<th>Operation300</th>
<th>T1</th>
<th>...</th>
<th>T300</th>
<th>Performance</th>
</tr>
</thead>
<tbody>
<tr>
<td>3699</td>
<td>2</td>
<td>...</td>
<td>3</td>
<td>77.69978</td>
<td>...</td>
<td></td>
<td>2841.763</td>
</tr>
<tr>
<td>1427</td>
<td>9</td>
<td>...</td>
<td>3</td>
<td>31.51063</td>
<td>...</td>
<td>320.6053</td>
<td>2779.744</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>...</td>
<td>3</td>
<td>366.2558</td>
<td></td>
<td>2721.452</td>
<td></td>
</tr>
<tr>
<td>3553</td>
<td>9</td>
<td>...</td>
<td>1</td>
<td>375.2200</td>
<td></td>
<td>2933.481</td>
<td></td>
</tr>
<tr>
<td>3591</td>
<td>9</td>
<td>...</td>
<td>3</td>
<td>71.8805</td>
<td>...</td>
<td>352.9384</td>
<td>2809.132</td>
</tr>
</tbody>
</table>
3.2 Methodologies

The feature selection technique and the discovery of tool patterns leading to a wafer lot showing low/high performances can be explained as follows:

**Step 1: Data preparation**

1.1 Remove irrelevant features, that are, lot # and timestamp ($T_1 - T_{300}$)
1.2 Replace missing value with a symbol ‘?’
   (Note that there are around 25% of missing values in several operational units.)
1.3 Sort data in ascending order according to the performance value (these data instances will be referred to by their numbers ranging from 1-4000)

**Step 2: Feature selection**

2.1 Set the threshold value $T$ as 0.60
2.2 Add a new feature, called class, with two distinct values: $c_0$ (low performance) and $c_1$ (high performance). The first half of instances (number 1-2000) are assigned class $c_0$ and the rest (instance number 2001-4000) are class $c_1$.
2.3 Prepare train and test data sets. Test set contains 200 data instances that are the first 100 instances and the last 100 instances (instances number 1-100 and 3901-4000). The train data set contains 1000 data instances that are instances number 101-600 and 3401-3900.
2.4 For all the available 300 features, do the analysis to justify appropriate training size by
   2.4.1 Dividing the train data into 20 subsets; each subset is an increment of 50 data instances (that is, 50, 100, 150, 200, 250, …, 950, 1000 instances)
   2.4.2 Then test predictive accuracy of each data subset using Naïve Bayes learning algorithm
2.5 Select a set of features that yield predictive accuracy greater than the threshold $T$

**Step 3: Perform data mining**

3.1 Use selected features that have been analysed by step 2 to extract features in both the train data and test data sets
3.2 Run selected data mining algorithm on the train data set and then test the accuracy of predictive model using the test data set

4 Experimentation and Discussion

4.1 Feature Selection Results

The step 2 in the previous methodologies is our main contribution. From the step 2.4 that is the analysis for a proper training size, we obtain the analysis result as shown in Figure 1. In the figure, Y-axis is predictive accuracy, whereas X-axis is the size of training data. Each scale along the X-axis is the power of 50. The best accuracy is at $X=7$, which is the data subset of size $7 \times 50 = 350$ data instances.

From the step 2.5, which is the selection of features giving accuracy higher than 0.6, we obtain the selection result as shown in Figure 2. With the accuracy threshold of 0.6, there are 7 features selected by the proposed method. These features are columns number 10, 85, 133, 141, 147, 198, and 261. Figure 3 shows the results of both proper training size and the best discriminative features.
4.2 Comparative Results of Performance Predictive Models

After selecting proper features and justifying appropriate training size, the prepared data are then used to induce a predictive model. We perform experimentation with seven model induction methods: decision tree induction (C4.5 algorithm), random forest (RF), naïve Bayes (NB), k-nearest neighbors (k-NN, with k=10), Adaboost (AdaB), artificial neural network (ANN, using voted perceptron algorithm), and support vector machine (SVM). All experimentations have been tested with the same data set. The predictive performances of the seven algorithms are summarized in Table 2 and graphically compared in Figures 4 and 5.

**Table 2. Predictive performance of each learning algorithm.**

<table>
<thead>
<tr>
<th>Learning algorithm</th>
<th>Use all features (300 operations)</th>
<th>Feature selection (7 operations)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Accuracy</td>
<td>Time (sec)</td>
</tr>
<tr>
<td>Decision Tree</td>
<td>64.00%</td>
<td>0.14</td>
</tr>
<tr>
<td>Random Forest</td>
<td>51.00%</td>
<td>0.14</td>
</tr>
<tr>
<td>Naive Bayes</td>
<td>54.00%</td>
<td>0.03</td>
</tr>
<tr>
<td>k-NN (k=10)</td>
<td>46.50%</td>
<td>1.17</td>
</tr>
<tr>
<td>AdaBoost</td>
<td>64.00%</td>
<td>0.15</td>
</tr>
<tr>
<td>ANN (Voted Perceptron)</td>
<td>51.50%</td>
<td>13.67</td>
</tr>
<tr>
<td>Support Vector Machine</td>
<td>50.00%</td>
<td>18.28</td>
</tr>
</tbody>
</table>
Operation 147

Tool #1
Tool #2
Tool #6
High Performance
Tool #7
Tool #3
Tool #4
Tool #5
Low Performance

Figure 6. A decision tree model to predict low/high performances of wafer lots.

It can be seen that our proposed feature selection technique can improve predictive performance, as well as can reduce training and testing time. The best predictive model in terms of accuracy is the k-NN when ten nearest neighbors are taken into account, but the decision tree model (as shown in Figure 6) is the most comprehensible model. Most user might consider that the performance predictive model (as low versus high performances of wafer lots at the end of the production line) in a form of decision tree conveys a very concise information that operational unit number 147 is a key factor for predicting low/high performance. Tools number 3, 4, and 5 applied in this operational unit result in a low performance wafer lots, whereas tools number 1, 2, 6, and 7 yield a high performance product.

5 Conclusion
We propose a novel technique for selecting only discriminative features and extracting appropriate amount of train data. The proposed technique is to help process engineers analyzing problematic tools and operational units in the semiconductor industries. Most semiconductor manufacturing is highly complex and has produced constantly hundreds of metrology data that are awaiting for process engineers to analyze for the purpose of maintaining efficient operations and getting optimum yield of high quality products. For such a large volume of measurement data, automatic data analysis technique such as data mining is essential.

The experimental results confirm efficiency of our feature selection technique. From the seven learning algorithms, six algorithms show significant improvement in terms of predictive accuracy and model induction time. The k-nearest neighbor model shows the highest improvement; predictive accuracy has been improved from 46.50% to 70.50%. But we found that the decision tree model is good at describing operational unit, which is annotated with specific tool number that has been applied in that unit. Such specific information is helpful for searching the root cause of problematic lots.

Acknowledgment
This work has been supported by grant from Suranaree University of Technology via the funding of Data Engineering Research Unit.

References:


