Detection of Faulty Semiconductor Wafers using Dynamic Growing Self Organizing Map

G. Russ and R. Kruse
Faculty of Informatics, University of Magdeburg, Germany

M. A. Karim[1], A. L. Hsu, S. Halgamuge and A. J. Smith
Department of Mechanical and Manufacturing Engineering,
University of Melbourne, Australia
([1] Corresponding author email: makarim@mame.mu.oz.au)

A. Islam
Department of Information Technology,
Espoo – Vantaa Institute of Technology, Finland

Abstract: **Solving product yield and quality problems in a manufacturing process is becoming increasingly more difficult. There are various types of failures and their causes have complex multi-factor interrelationships. Semiconductor manufacturing is very complex due to the large number of processes, diverse equipment set and nonlinear process flows. Its manufacturing database comprises of hundreds of process control, process step and wafer probe data. This huge volume of data coupled with quicker time to market expectations is making finding and resolving problems quickly an overwhelming task. In this study, a methodology developed using dynamic growing self-organizing map (GSOM) to detect the faulty products in a wafer manufacturing process. As part of the methodology, a clustering quality measure was developed to evaluate the performance of the algorithm in separating good and faulty products. Results show that the algorithm was able to separate good and faulty products from the raw data. Even though this work has focused mainly on clustering good and faulty products, the technique can be extended to model the failure causes of the lower yielding products.**

I. INTRODUCTION

Manufacturers constantly face the yield and quality problems as they constantly redesign their processes for the rapid introduction of new products and adoption of new process technologies. Automatic, real-time collection of manufacturing data is common practice for modern manufacturing processes. The analysis of large volumes of manufacturing data, interpreting results, and implementing design improvements is computationally intensive and time-consuming. A high priority goal for wafer manufacturing is finding the most probable causative factor(s) that discriminate between low yield and high yield product by quickly examining the historical manufacturing data.

In order to decrease design cycles and, thereby, the time to market for new products, it is important to have a method for quickly and efficiently analyzing manufacturing data, predicting the effects of design changes, and determining the best design parameters. Advances in data mining have provided techniques for automatically discovering underlying knowledge from large amounts of data.

The objective of the present research was to develop a methodology using GSOM [1] to detect faulty products from manufacturing data and to demonstrate the effectiveness of the developed methodology by applying in a semiconductor wafer manufacturing. In this paper, the results of an industrial application of the methodology are presented. The methodology was successfully implemented in separating good and faulty products in a wafer manufacturing facility.

II. DATA MINING APPROACHES

Data mining methods can be used to extract useful information from enormous volumes of data. It enables trends to be discovered from observed data, and has the potential for providing the underpinning technology for decision support tools. Such methods may use machine learning, statistical methods, and visualization techniques to discover and present knowledge in a form which is easily comprehensible to humans.

Self-Organising Maps (SOM) [1, 2] has been widely used in data mining/knowledge exploration to visualise high-dimensional data and to reduce its dimensions. Many variants extending the conventional SOM's capabilities were proposed to allow more flexibility and adaptiveness by introducing controllable and consecutive growth during the training process. The next section will briefly explain the SOM and GSOM algorithms.

III. SOM and GSOM ALGORITHMS

A. Self-Organising Map

For a self-organising map of k neurons, where i ≤ k, for each input vector x, there exists a winning neuron $\mathbf{m_b}$ with minimum euclidian distance to the input x, such that

$$\|x - m_b\| = \min_i \{\|x - m_i\|\}. \quad (1)$$

This is referred to as the competitive learning part of the learning process, since all neurons compete to be the winner.

Then the winning neuron's and its neighbouring neurons' weights at time t+1 are adapted according to the learning rule

$$m_i(t+1) = m_i(t) + \alpha(t) h_{bi}(t)[x - m_i(t)], \quad (2)$$

where, α(t) is a monotonically decreasing learning rate and $h_{bi}(t)$ is a Gaussian neighbourhood kernel given by

$$h_{bi}(t) = e^{-\frac{\|r_b - r_i\|^2}{2\sigma^2(t)}}, \quad (3)$$

where $\mathbf{r}_b$ and $\mathbf{r}_i$ are positions of the winning neuron and neuron *i* respectively, and σ(t) the neighbourhood radius. The learning rule makes the weights of winning neuron and its neighbours more similar to the input x.

Self-organising maps have the ability to compress data, deal with missing data, preserve data similarity and represent data in a visible low-dimensional space. However, existing self-organising map algorithms suffer from drawbacks like static structure, border effect, structure warping, sensitive to learning rate parameter, inexistence of cooperative and competitive learning, inability to fully preserve data similarity and all without ability to apply to time series data [3, 4].

B. GSOM and Modified GSOM (with hexagonal topology)

To overcome some of the known limitations of SOM, Alahakoon et al.[1] proposed a dynamic growing self-organizing map (GSOM) algorithm. Hsu et al. [5] further improved GSOM with a hexagonal topology. The original GSOM only supports rectangular topology. Hexagonal topology is known to have better topology preservation for SOM [2]. GSOM has the same topology structure and weight vector adaptation rules as SOM. A significant difference between these two algorithms is that SOM is not capable to grow but GSOM grows according to its own growing criterion. A parameter of growth, growth threshold (GT), is defined as:

$$GT = -D \times \ln(SF), \quad (4)$$

where *D* is the dimensionality of data and SF is the user defined spread factor that takes values between 0 and 1with 0 representing minimum and 1 representing maximum growth.

The algorithm first identifies the winning node. Then an accumulated error *E* of the winning node is updated by the following rule:

$$E(t+1) = E(t) + I - w_{winner}, \quad (5)$$

where *I* is the input vector and w_{winner} is the weight vector of the winning node. If the winning node is the boundary node and *E* exceeds GT, growing is initiated on that node to fill the surrounding unoccupied spaces of the lattice. If *E* of the winning node exceeds GT but the winning node is not a boundary node, *E* is propagated outwards to other neighbouring nodes. Weights of the new nodes will be initialized according to the following equation:

$$w_{new} = 2w_{winner} - w_{opposite}, \quad (6)$$

where w_{opposite} represents the weight of the node topologically opposite to the new node. If these are not the topologically opposite nodes, weights of the new nodes will be calculated according to the following equation:

$$w_{new} = w_{winner} + w_{other1} - w_{other2}, \quad (7)$$

where w_{other1} and w_{other2} are weights of the nodes nearest to the new node. For hexagonal topology, equation (6) is applicable as there will always be a neighbour of the winning node that is topologically on the opposite side of the new node.

IV. DETECTING FAULTY WAFERS WITH MODIFIED GSOM

Dataset used in this study for experimental purpose was made available by Motorola USA. The quality problem of Motorola wafer fabrication is described below [6].

A. The problem

Motorola wafer manufacturing facility had a stable and mature fabrication facility and was consistently achieving very high product yield. However, for unknown reasons, the average product yields started to degrade by certain percentage. Moreover, the degradation was not continuous, rather periodic, for short period of time. In manufacturing process, periodic degradations are more difficult to detect than continuous failures. Yield improvements over such narrow ranges at already high yield are also difficult. Another problem was that, Motorola's established process control measurements were not sensitive enough to detect this particular problem during manufacturing. Only the final electrical testing was able to detect the problem. The cause of the failure was the excessive transistor collector/emitter leakage current. Unfortunately, when the problem was detected, substantial value had been added to the faulty wafers already.

B. Application of Standard fault detection methods

More than five years was spent to determine the reasons for this periodic failures applying standard technique called a "Design of Experiments" (D.O.E)[6]. Manufacturing and quality control engineers proposed several models to explain the leakage current occurrence. Device, diffusion and epitaxial silicon growth engineers performed over 30 design of experiments (DOE) throughout this time period to address every possible cause they thought. However, no significant progress was made. After five years, a special cross-functional team was formed to address the leakage current problem. For about a year, the team performed different types of experiments to determine the cause of leakage. Finally, the designed experiments were able to detect the cause of the problem. It was found that a collector to emitter pipe caused by a defect stacking fault was occurring somewhere in the process. It took well over five years to find it.

C. Data mining approach

The authors believed that the Motorola's wafer yield problem described above can be solved using modified GSOM algorithm. In the context of analysis of manufacturing quality problems, the focus involves two main aspects- separation of good and faulty products and identifying the reason for yield failure. The present study focused in to the first aspect. The challenge is not solely in the clustering alone, but also to obtain meaningful and adequate number of clusters. With meaningful clusters, grouped in appropriate numbers, identification of the reasons that contribute significantly to the differentiation of clusters would become a simpler task. We apply a fully unsupervised methodology that uses a combination of dynamic SOM tree [11] and Growing Self-Organizing Maps (GSOMs) in manufacturing data analysis.

D. A quality measure

It was anticipated that 100% separation of good and faulty products was not possible. This is because there are so many attributes and there could be many reasons for yield failure. In order to compare the results of running simulations with different parameters on the same dataset, or with the same parameters on differently preprocessed data, this section will introduce a quality measure which takes the complete cluster map into account and can be generated automatically as an objective quantifier.

The desired clustering would be a binary clustering at best, i.e., the complete separation of the 'good' from the 'faulty' products on the GSOM. The data available for developing the benchmarking method are (a) the original distribution of data between 'good' and 'faulty', and (b) the data from the generated map, i.e., the distribution of input vector references among the map's neurons.

The benchmark should yield a number between 0 and 1 as a quality measure of the input vector separation among the neurons. Zero (0) means no difference in data distribution, i.e. no clustering at all and 1 means a high probability of good clustering.

Mathematically, clustering quality can be expressed as:

$$CQ = \sum_i \max\left\{ \frac{\frac{g_i}{n_i} - \frac{G}{N}}{1 - \frac{G}{N}} * \frac{n_i}{N}, 0 \right\} + \sum_i \max\left\{ \frac{\frac{b_i}{n_i} - \frac{B}{N}}{1 - \frac{B}{N}} * \frac{n_i}{N}, 0 \right\}, \quad (8)$$

where

B = *total number of faulty products,*

G =total number of good products,

N =total number products (B+G),

b_i = number of faulty products in neuron i,

g_i = number of good products in neuron i, and

n_i = number of products in neuron i.

A perfect separation of input vectors is achieved if all the neurons contain only input vectors from one class (i.e. either 'good' or 'faulty'), in which case the benchmark should yield 1. A bad separation (or no separation, i.e. cluster neurons have same distribution of original dataset) should be benchmarked with a 0. For example, if we consider 100 inputs and 80 of them are good and 20 are faulty. Let us consider that final cluster has 3 neurons and these three contain 40 good, 20 faulty and 40 good products respectively. If we use these values in equation (8), CQ will be 1. The detail of this example is presented in Table 1.

TABLE 1 : Sample calculation of CQ

Total inputs	Neuron Index	Inputs in each neuron		CQ of neuron	CQ of the cluster
		Good	Faulty		
100	1	40	0	0.4	1.0
	2	40	0	0.4	
	3	0	20	0.2	

The proposed quality measure was evaluated using Motorola wafer yield data. Two test datasets were generated to evaluate the effectiveness of the developed clustering quality measure (and, simultaneously, the functionality of the used GSOM implementation as well). As the interests lie in assessing binary clustering quality, two samples which represent the original dataset's two different classes of the original dataset (good and faulty) were taken from the normalised dataset and repeated 50 and 500 times respectively. The resulting test datasets had dimensions of 59x100 and 59x1000 and, due to the generation, consist of two easily distinguishable classes. These two datasets were then fed into the GSOM.

The generated GSOM maps can be found in Figure 1. Both maps were computed with the parameter set [SF=0.5, GP=3] and showed the desired CQ value of 1 since both classes are perfectly separated on the maps.

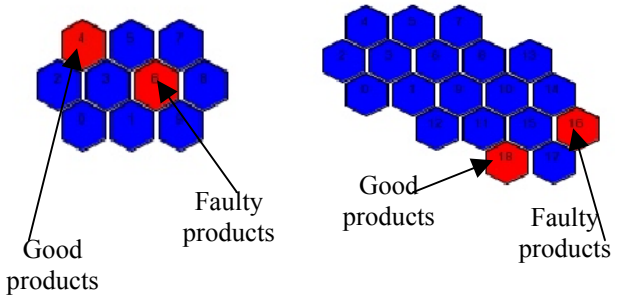

Figure 1: Generated maps with test dataset, (a): 59x100, (b): 59x1000

E. Data Preprocessing

Data preparation (or problem reformulation) is an essential step in the data mining process. The original manufacturing data (such as machine ID) cannot be directly utilized by GSOM. Therefore the data must be reformulated in terms that can be handled by the data-mining algorithm. Categorical data should be converted to numerical values for the GSOM.

The original dataset was historical wafer data collected for 2500 wafers from a 2-month period. The input database measured 133 parameters by 17,246 entries organized into an Excel file. The data consisted of:

- Wafer Probe Data - The pass/fail count per wafer of 39 wafer probe functional tests.
- Process Control Data - There were 59 numerical electrical PC measurements probed at 8 sites per wafer.
- Process Step Data – There were 33 process parameters, hand collected parametric and non-parametric data such as material vendor/lot, wafer boat position, operator & machine IDs, etc.

In the dataset provided, each entry had a reference number which determined whether the product was a good one or a faulty one. Therefore number of good and faulty products was known, allowing us to compare with the results.

1). Expansion of categorical data: The dataset consisted of numerical as well as categorical data. Categorical values were converted to 0 and 1 as explained in Table 2. In this process each column with categorical data was expanded to many columns depending on the number of different categorical values in that column.

TABLE 2: Expansion of categorical data

Original data		Expanded data			
X103		***P4812***	***P4815***	***P4753***	***P4683***
P4812	⇒	1	0	0	0
P4815		0	1	0	0
P4812		1	0	0	0
P4753		0	0	1	0
P4683		0	0	0	1

2). Removal of outliers: Among the original dataset there are few outliers which are most probably caused by incorrect measurements from the production lines (e.g. faulty meters or errors during transmission); which manifest by being several orders of magnitude different from the remaining majority of the data. Assuming that the normalisation is carried out in the usual way, even a single outlier can influence the data mining process. A more intuitive explanation can be obtained from Figures 2, where the immediate effect of the outliers is obvious, scaling the majority of values to nearly zero and the few outliers to one, therefore rendering the data mining process nearly useless. However, it shall be noted that the distribution of 'good' and 'faulty' input vector references did not undergo a major change after outliers were eliminated; the percentage of 'good' products changed from 85.92% to 86.04% whereas the percentage of faulty products was reduced by the respective amount.

Figure 2: Data distribution after normalisation including outliers

3). Normalization: After removing the outliers, every data attribute was scaled between 0 and 1 using Equation (9).

$$N_i = \frac{X_i - X_{\min}}{X_{\max} - X_{\min}}, \qquad (9)$$

where,

$N_i =$ Normalized value of i^{th} data,

$X_i =$ Input value,

$X_{\max} =$ Maximum value of each column, and

$X_{\min} =$ Minimum value of each column

F. Simulation Results

Clustering quality was determined for different spread factor (SF) and number of growing phases (GP). Spread factor determines the final size of the map by providing the user with a convenient, high-level way to influence the map's internal growth accordingly [1]. A larger spread factor means a more spread-out and therefore larger final GSOM. SF was varied

from 0.1 to 0.995 during simulations. GP determines the running time of the algorithm and also the size of the map. It was varied from 1 to 100. However, GP more than 10 gave no better results but increased computational time significantly. Simulations were run with data with and without eliminating the 'outliers'. Different parameters in the programme, namely spread factor and number of growing phase, were changed to check the improvement in cluster quality. The dataset before eliminating the outliers was termed as 'old' and after eliminating outliers was termed as 'new'.

Figure 3 shows the simulation results of the old dataset for different spread factor and number of growing phases. It can be seen that clustering quality improved with the increase of both SF and GP. Our experiments show that GP>10 do not improve results significantly. Hence the plots in Figures 3-5 are limited to GP<10.

Similarly, simulation was run with the 'new' dataset. Figure 4 shows the simulation results with new dataset. A comparison between the results with old and new dataset was made and presented in Figure 5. It can be seen that after eliminating outliers from the data, CQ has risen by 3- 12% (SF 0.8) and 2-8% (SF 0.9). However, overall CQ peaks at about 0.5 when perfect clustering quality should be close to 1.

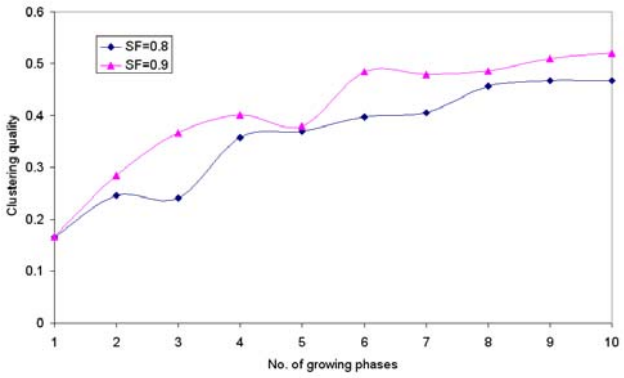

Figure 3. Clustering quality of simulation using old data

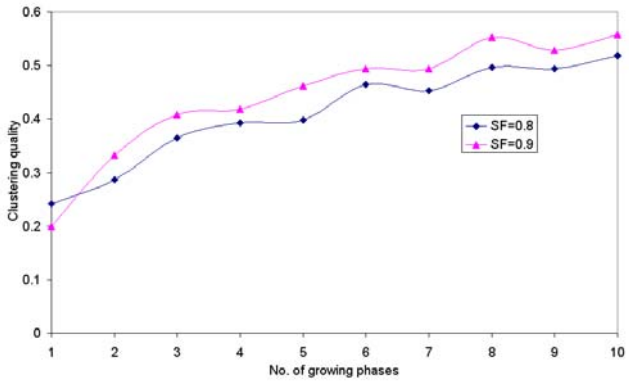

Figure 4. Clustering quality of simulation using new data

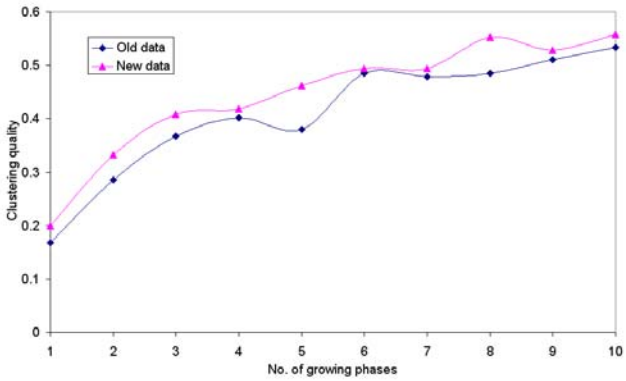

Figure 5: Comparison of CQ with old and new data (SF = 0.9)

It is thought that introduction of a large number of attributes (because of the expansion of categorical data) created unnecessary noises in the dataset. It may be necessary and of great insight to study the effect of removing several attributes from the input dataset and check its impact on the generated maps. This will generate and/or confirm knowledge about possible non-contributing attributes.

It should be noted here that CQ is the measure of all the neurons in a cluster. In practice, it is not possible to have all neurons with 100% good or faulty products. Although CQ is in the range of 0.55, the GSOM was able to make few clearly different clusters for good and faulty products. Figure 6 shows the clusters for new data (with SF=0.9 and GP=10). In this cluster, many nodes were found with 100% good or 100% faulty products. It shows that the GSOM was able to separate good and faulty products. Moreover, clusters with good and faulty products were found at opposite ends. This indicates that distance between good and faulty products was correctly identified. As there are neurons with 100% good and faulty products, it is possible to find the underlying reason for this separation.

For manufacturing data, the objective is not limited to finding the separation of good and faulty products. The main objective is to find the underlying reason for poor yield. To discover this, it is necessary to create cluster of good and faulty products, as has been done in this study, and then apply rule induction method (to be developed under this project) to find the reasons for making two clusters. The technique is being extended to model the failure causes of the lower yielding products.

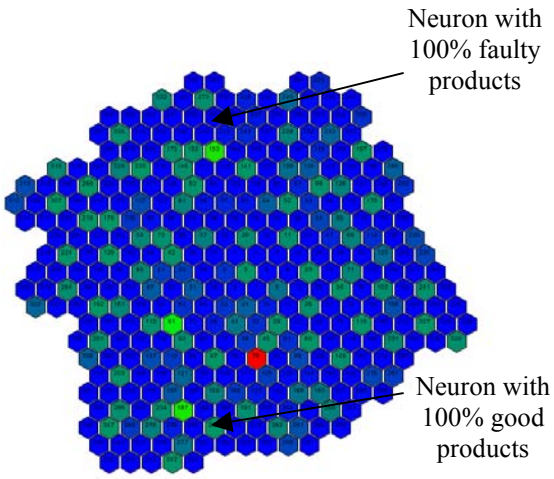

Figure 6: Cluster map for new data (with SF=0.9 and GP=10)

V. CONCLUSIONS

This paper has described an intelligent data mining system for analysis of wafer manufacturing data. The main contributions of this paper include

1. Partial Automation of knowledge discovery in the failure data;
2. Applying of GSOM algorithm to automatically analyze the raw manufacturing data to separate good and faulty products.

The preliminary experiments with data mining and wafer manufacturing data show that it is possible to separate good and faulty products using manufacturing data, hence the causes of low yielding wafers. The learning approach in the system can significantly reduce product design cycle time both by increasing the level of automation in the failure analysis and by offering insights that may not be obvious to the design engineer. Our preliminary studies with GSOM for failure analysis indicate that the proposed data mining approach is flexible and may be applied to a number of other design and manufacturing processes. Future work will likely include further automation of the steps for "rule induction," to identify underlying reasons of separating good and faulty products. Also an appropriate technique to preprocess categorical data may lead to improved system performance and prediction accuracy.

ACKNOWLEDGEMENT

The authors like to thank Robert M. Gardner, Director Intelligent Systems Research Lab, Motorola, USA for kindly providing the valuable semiconductor wafer data.

The Authors also like to thank Prof. Malcolm Good and The Research Centre for Advanced By-Wire Technologies (RABiT) for allowing to use their excellent computing facilities.

Partial funding from the Australian Research Council (ARC) is appreciated.

REFERENCES

[1] D. Alahakoon, S.K. Halgamuge, and B. Srinivasan, "Dynamic self-organising maps with controlled growth for knowledge discovery". *IEEE Transactions on Neural Networks, Special Issue on Knowledge Discovery and Data Minin*g, 2000. 11(3): p. 601-614.

[2] T. Kohonen. *Self-Organising Maps*. 1997: Springer Berlin.

[3] R. M. Miller, *Computer Aided Financial Analysis*. 1990: Addison-Wesley Publishing Company.

[4] E. Gately, *Neural Networks for Financial Forecasting*. 1996: John Wiley & Sons.

[5] A.L. Hsu, S.-L. Tang, and S.K. Halgamuge, "An unsupervised hierarchical dynamic self-organizing approach to cancer class discovery and marker gene identification in microarray data". *Bioinformatics*, 2003. 19(6): p. 2131–2140.

[6] R.M. Gardner, J. Bieker, and S. Elwell. "Solving Tough Semiconductor Manufacturing Problems using Data Mining". *Advanced Semiconductor Manufacturing Conference and Workshop*, 2000 IEEE/SEMI 12-14 Sept. 2000 Page(s):46 - 55.