「정보시스템연구」 제19권 제1호
한국정보시스템학회
2010년 3월, pp. 19~33

# An intelligent system for semiconductor yield classification with soft computing techniques*

Jang Hee Lee** · Sung Ho Ha***

〈목 차〉



## Ⅰ. Introduction

A yield is an important index to measure the process quality, which signifies the fraction of total input transformed into shippable output. To prevent low yield and maintain high yield is crucial to the success of the semiconductor industry. However, controlling yield is difficult due to many process parameters affecting yield variation and complex relations. For efficient yield monitoring, most manufacturers have tried to discover exact relations between yield and process parameters. In this study, we devise a decision support system with a more accurate yield prediction method, called an intelligent yield prediction system (IYPS) to detect high and low yields. The IYPS adopts a self-organizing map (SOM) neural network to identify patterns of process parameters, and predicts a yield level of a new manufacturing lot by performing feature-weighted case-based reasoning (CBR) which is based on the weights of the attributes

* 이 논문은 2008년도 한국기술교육대학교 연구제 파견연구비 지원에 의하여 연구되었음.
** 한국기술교육대학교 산업경영학부 부교수, janghlee@kut.ac.kr
*** 경북대학교 경영학부 부교수(교신저자), hsh@mail.knu.ac.kr

calculated from the trained back-propagation neural networks (BPN).

## Ⅱ. Related works

When looking through the academic journals related to semiconductor (Tsay, 2006), data mining tools can be used to discover the relations between the yield and process parameters and to predict the yield of a lot based on the relations (Shapiro *et al.*, 1996). Yield management in semiconductor manufacturing is complex and difficult, due to many interrelated factors that affect it. Therefore hybrid approaches of several mining tools have been used in several studies to improve yield prediction accuracy (Kang *et al.*, 1998; Shin and Park, 1999; Shin *et al.*, 2000). Kang *et al.* (1998) showed an effective hybrid application of mining tools such as inductive decision trees, BPN, and SOM neural network for controlling semiconductor yield. Shin and Park (1999) and Shin *et al.* (2000) used hybrid approaches of neural networks and CBR to predict semiconductor yield.

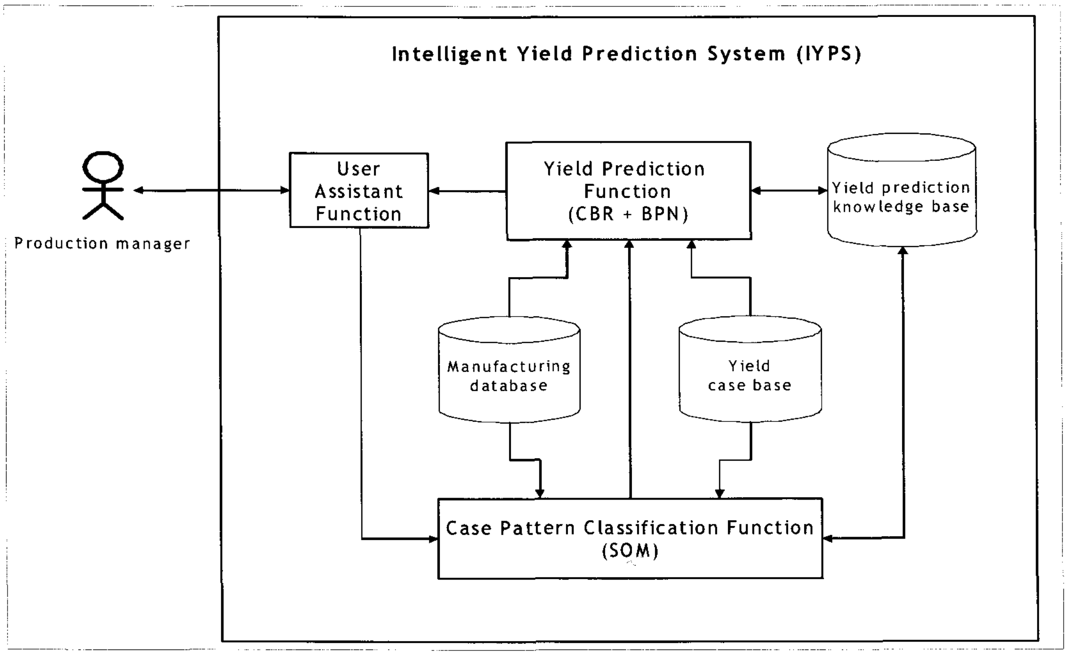
Many other studies using a hybrid approach to analyze the causes of defects related to semiconductor yield, on the other hand, have been reported (Chien *et al.*, 2007; Hsu and Chien, 2007; Li and Huang, 2009). Chien *et al.* (2007) applied a decision tree and *k*-means clustering to finding out the significant causes of faults and the variations in production processes that generate yield loss. Hsu and Chien (2007) proposed an integration of spatial statistics and adaptive resonance theory (ART) neural networks for interpreting patterns of wafer bin maps associated with manufacturing defects. Li and Huang (2009) integrated a SOM neural network and Support Vector Machine to analyze wafer bin map data related to manufacturing defects.

Based on those hybrid approaches, this study considers using a hybrid CBR, which is a kind of instance-based approach to store the presented training data (Kim, 1999; Roh *et al.*, 2005). According to the survey conducted by Liao (2005), the applications implementing hybrid CBRs have mainly classified into four areas: manufacturing design, fault diagnosis, knowledge modeling and management, and medical planning and application. Among these applications, the hybrid CBR approach has been extensively adopted in manufacturing design and fault diagnosis. Liao (2004) integrated a CBR method with a multi-layer perceptron for the identification of failure mechanisms in the failure analysis process. Yang *et al.* (2004) integrated CBR with an ART-Kohonen neural network to enhance fault diagnosis of electric motors. Tan *et al.* (2006) integrated CBR and fuzzy ARTMAP neural network to make optimal manufacturing technology investment decisions. Saridakis and Dentsoras (2007) introduced a case-based design with a soft computing system to evaluate the parametric design of an oscillating conveyor.

Some previous studies have reported that a hybrid use of CBR has shown better performance to predict the yield of the semiconductor (Kang *et al.*, 1998; Shin and Park, 1999; Shin *et al.*, 2000). For the advanced prediction of semiconductor yield, we focus on a feature-weighted CBR, which combines CBR and BPN together (Wettschereck and Aha, 1995; Hastie and Tibshirani, 1996; Wettschereck *et al.*, 1997; Shin *et al.*, 2000; Ahn *et al.*, 2006; Chien *et al.*, 2007; Hsu and Chien, 2007; Wang, 2008; Li and Huang, 2009). The feature-weighted CBR uses the weights of the attributes calculated from the trained BPN. When a new, unseen query instance is encountered in CBR, a set of similar instances are retrieved from a case base and are used to predict a target value for the new instance. The *k*-Nearest Neighbor (*k*-NN) method has been widely used for retrieving the *k*-nearest cases to estimate the target value of a new query. It generally calculates the distance between cases by allowing equal importance to all attributes of case, which is unfortunately not true in many practical applications. Owing to this drawback, many variants of *k*-NN have been proposed to assign larger weights to the more relevant attributes in the case retrieval, which is the feature-weighted CBR.

# Ⅲ. Intelligent yield prediction system (IYPS)

Figure 1 presents a framework of an intelligent yield prediction system (IYPS) based on data




<Figure 1> Architecture of Intelligent Yield Prediction System (IYPS)

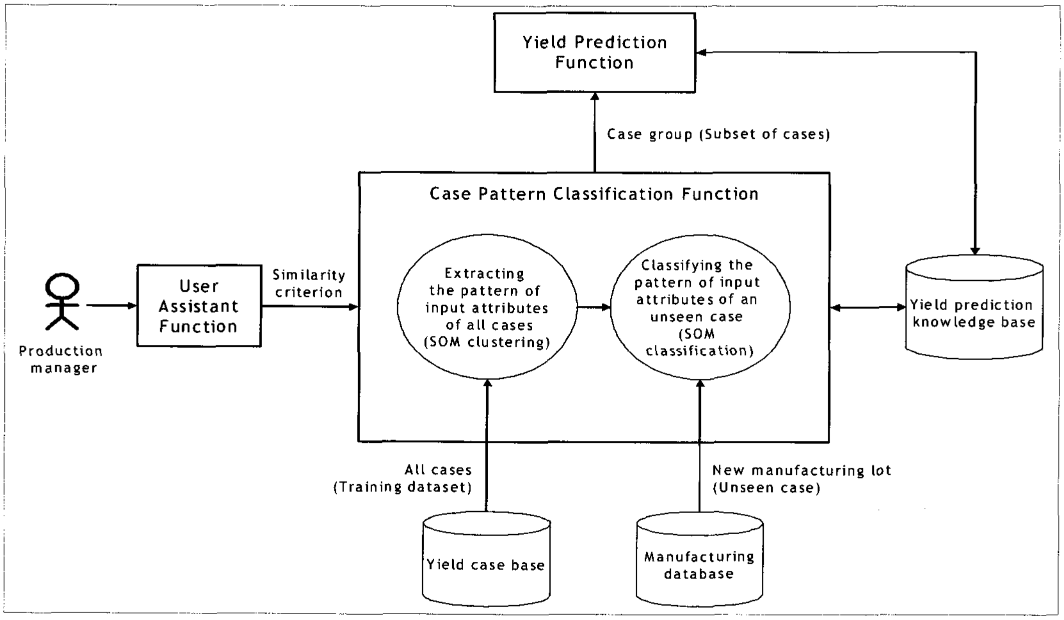
mining technologies.

The IYPS has three functions: case pattern classification (CPC), yield prediction (YP), and user assistant (UA) functions. The CPC function is in charge of pattern classification of attributes of a new manufacturing lot. The YP function takes a role of predicting a yield level of the new lot by using a feature-weighted CBR. The UA function acts as an interface between the IYPS and users, including production engineers. It relays user requests to the CPC and enables the production managers to monitor the status and results of the yield prediction process of the YP.

## 3.1 Case Pattern Classification (CPC) function

The CPC utilizes a SOM to extract distinctive patterns from all the cases residing in the case base, divides cases into several homogeneous groups, and classifies the patterns of a new unseen case (see Figure 2). The reasons for clustering is that the CBR with all training data may even hinder an accurate prediction. Thus, the CPC selects homogeneous cases and then, supplies them to the YP function instead of providing all cases.

We employ a SOM for clustering, because of the SOM's good performance on the feature extraction and pattern recognition. Many existing studies of SOM have reported that it has shown an excellent performance in various engineering problems, including clustering, pattern recognition, dimension reduction, feature extraction, and dependencies discovery and abstractions from raw data (Marks and Goser,



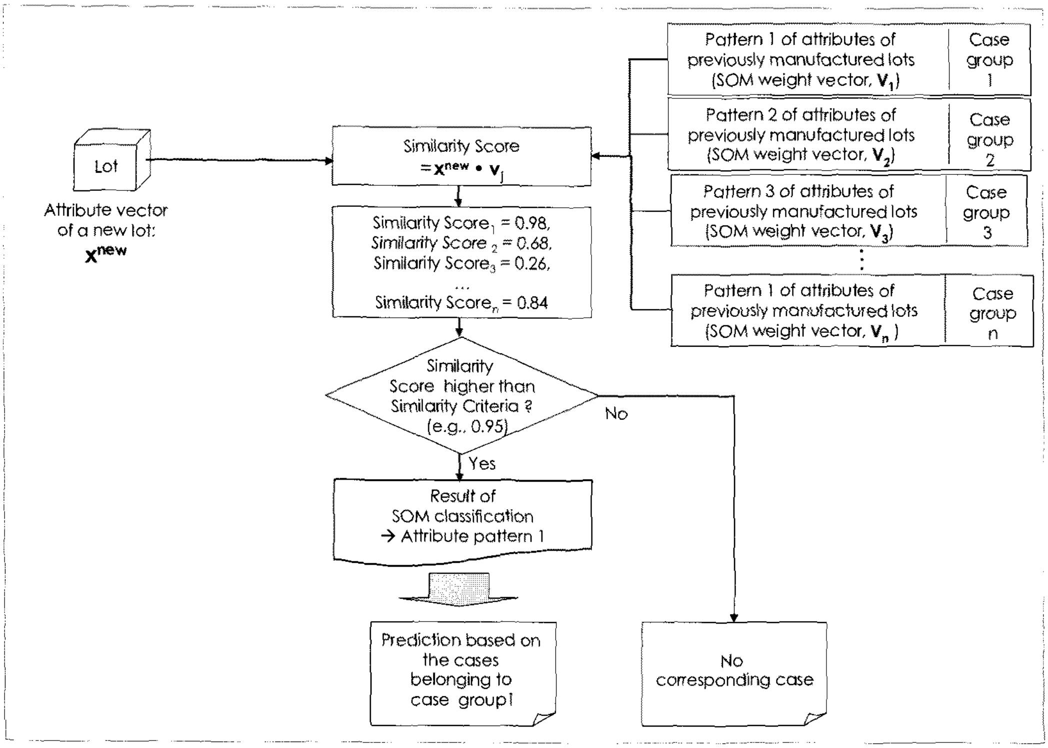

<Figure 2> CPC function and its interaction with other functions

1988; Caviglia *et al.*, 1990; Hemani and Postula, 1990; Kim and Kyung, 1991; Jiang and Jabri, 1992; Kiziloglu *et al.*, 1993; Kohonen *et al.*, 1996; Kaski, 1997; Lee *et al.*, 2001).

Given a case base ***L***, each case $\overrightarrow{x_i}=(x_{i1},x_{i2},\ldots,x_{in},x_{ic})$ is defined by a set of $n$ features, where $x_c$ is a target class value, such as high or low yield. The SOM algorithm trains all $\overrightarrow{x_i^{old}}=(x_{i1}^{old},x_{i2}^{old},\ldots,x_{in}^{old})$ in the case base and extracts significant feature patterns from all cases. The extracted patterns are stored in the yield prediction knowledge base with the form of a SOM weight vector, $\overrightarrow{v_j}$

When the attributes of a new lot $\overrightarrow{x^{new}}=(x_1^{new},x_2^{new},\ldots,x_n^{new})$ are gathered, the CPC classifies the $\overrightarrow{x^{new}}$ into one of the SOM weight vectors (i.e., feature patterns of the previously manufactured lots) through the SOM classification process. The SOM classification produces a *similarity score* by calculating the inner product between SOM weight vectors ($\overrightarrow{v_j}$) and an attribute vector of a new lot ($\overrightarrow{x^{new}}$). The similarity score indicates the level of similarity between these two vectors.

When a similarity score is larger than a *similarity criterion* that is determined by the production manager who continues to control a product yield, a new lot is decided to be classified




<Figure 3> New manufacturing lot classification performed by the CPC

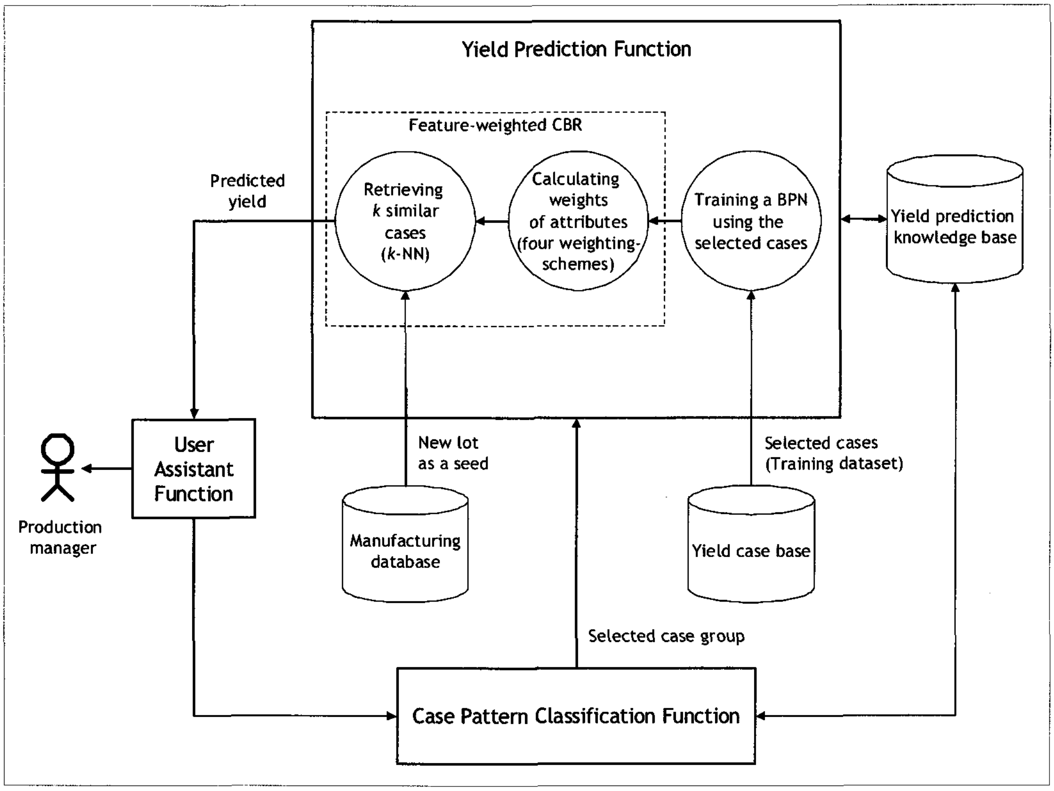
into the pattern of the corresponding SOM weight vector and it is possible to say that the new lot has a pattern of yield levels of cases belonging to the weight vector (Lee *et al.*, 2001). Figure 3 shows the process of classifying an attribute vector of a new manufacturing lot $\overrightarrow{x^{new}}$ into the case group 1 (i.e., $\overrightarrow{v_1}$), since the similarity score is largest (0.98) and is higher than the predefined similarity criterion (0.95) as well.

Then, the CPC supplies the YP function with the selected cases belonging to the corresponding case group 1. When a new lot is not classified into any of these case groups, the CPC transmits a "No corresponding cases" message to the YP and adds the new lot to the case base as a new one, after gathering its yield value.

## 3.2 Yield Prediction (YP) function

The YP calculates the yield level of each new lot by adopting a feature-weighted CBR, which combines CBR and BPN. Some previous studies have reported that the feature-weighted CBR has successfully predicted semiconductor yield (Kang *et al.*, 1998; Shin and Park, 1999; Shin *et al.*, 2000; Im and Park, 2007).

Whenever the CPC requests yield prediction, the YP trains a BPN along with the selected case




<Figure 4> Yield prediction process conducted by the YP

group and extracts attribute weights according to four weighting schemes (Sensitivity, Activity, Salience, and Relevance) to improve the retrieval accuracy of CBR. Notice that the selected case group contains most similar cases to the new lot within the yield case base. The YP generates a predicted yield rate for a new lot through the following steps (see Figure 4):

**(1) Step 1: To train a BPN along with the selected cases**

The YP trains a BPN by inputting selected cases, $\vec{x_i}$, belonging to the case group which is provided by the CPC. In this study, the YP adopts a BPN architecture of (*n-M-2*), which indicates *n* process parameters, *M* hidden nodes, and *2* yield levels, such as high-yield and low-yield levels. Through the training, the BPN keeps the intrinsic nature of the training cases on the connection weights between nodes.

**(2) Step 2: To calculate attribute weights using four weighting methods**

The YP extracts a set of feature weights from the trained BPN by employing four feature-weighting methods: Sensitivity, Activity, Saliency, and Relevance. All of the weighting methods utilize the connection weights and activation patterns of nodes in the trained BPN to calculate the importance of each attribute (Wettschereck *et al.*, 1997; Liu and Motoda, 1998; Zhang *et al.*, 2002). The *Sensitivity* weighting method calculates the sensitivity of an input node, which is calculated by removing the input node from the trained BPN. The *Activity* weighting method calculates the activity of a node, which is measured by its variance of activation level from training case. The *Saliency* weighting method

<Table 1> Four feature-weighting methods

| Weighting method | Equation |
|---|---|
| Sensitivity ($Sen_i$) | $Sen_i = \frac{\sum_{CG} \lvert P^0 - P^i \rvert}{k}$ (1) |
| Activity ($Act_i^{input}$) | $Act_i^{input} = var(x_i)\sum_{j=1}^{M}\left((w_{ji}^{(1)})^2 Act_j^{hidden}\right)$ (2-1)<br>$Act_j^{hidden} = (w_j^{(2)})^2 var\left(\sigma\left(\sum w_{ji}^{(1)} x_i\right)\right)$ (2-2) |
| Saliency ($Sal_i$) | $Sal_i = \sum_{j=1}^{M}\left((w_{ji}^{(1)})^2 (w_j^{(2)})^2\right)$ (3) |
| Relevance ($Rel_i^{input}$) | $Rel_j^{hidden} = (w_j^{(2)})^2 var(w_{ji}^{(1)})$ (4-1)<br>$Rel_j^{input} = \sum_{j=1}^{M}\left((w_{ji}^{(1)})^2 Rel_j^{hidden}\right)$ (4-2) |

calculates the saliency of a node, which is proportional to the square of its connection weights in the trained BPN. The *Relevance* weighting method calculates the relevance of a node, which is measured by the variance of connection weights in the trained BPN.

Table 1 summarizes each of four feature-weighting methods. Note that $p^0$ is the predicted value when an attribute $i$ is left and $p^i$ is the predicted value when the attribute $i$ is removed. $CG$ is a set of training cases within a case group and $k$ is the number of training cases. $var()$ is the variance function and $\sigma()$ is the activation function. $A_j^{hidden}$ is the activity of a hidden node $j$, $w_{ji}^{(1)}$ is the connection weights from the input node $i$ to the hidden node $j$, and $w_j^{(2)}$ is the connection weights of hidden node $j$ in the trained BPN. $R_j^{hidden}$ is the relevance of a hidden node $j$.

**(3) Step 3: To predict yields through using a k-Nearest Neighbor method**

The YP applies a $k$-NN method along with the Sensitivity, Activity, Saliency, or Relevance weight to retrieving the $k$ most similar cases from the case group, and then, determines a weighted value of output class as a predicted yield value for a query. It calculates the distance between a query $\overrightarrow{x^q}$ and a case $\overrightarrow{x^{CG}}$ by using the following equation:

$$Distance\left(\overrightarrow{x^q}, \overrightarrow{x^{CG}}\right) = \sqrt{\sum_i^n w_i \times difference\left(x_i^q, x_i^{CG}\right)^2} \tag{5}$$

where $difference\left(x_i^q, x_i^{CG}\right)$ becomes $\left|x_i^q - x_i^{CG}\right|$ when $x$ is numeric, 0 when $x$ is symbolic and $x_i^q = x_i^{CG}$, or 1 otherwise. $w_i$ is one of the Sensitivity, Activity, Saliency, or Relevance weight of the $i$th input attribute, and $x_i^q$, $x_i^{CG}$ is the $i$th attribute values of $\overrightarrow{x^q}$, $\overrightarrow{x^{CG}}$.

### 3.3 User Assistant (UA) function

The UA function accomplishes the particular user's interaction with the CPC or YP function. It passes user requests to the CPC and enables users to see execution results from the YP. To this end, the UA has multimedia user interface components and has operational facilities components. The multimedia user interface components manage interactions with users, and the operational facilities components provide a multimedia presentation to the user.

## Ⅳ. Verification of the intelligent yield prediction system

For the verification, we have implemented a web-based prototype system of IYPS, which has been developed primarily using the web program language, JSP. It had three functional modules, including case pattern classification (CPC), yield prediction (YP), and user assistant (UA) modules. We applied it to the real data of a semiconductor manufacturing company in Korea. Because of its data security enforcement, only 454 lot data of a semiconductor product were collected, which consisted of 227 high-yield and 227 low-yield lots. Sixteen process variables, such as critical dimension, film thickness, uniformity, and yield level were allowed to be collected.

The yield levels of 90% or more and 60% or less correspond to the high-yield and low-yield levels, respectively in that company. In addition, 454 lots were divided into 270 training data (60%) and 184 testing data (40%). The 270 training lots consisted of 135 high-yield and 135 low-yield lots. The 184 testing lots were comprised of 92 high-yield and 92 low-yield lots.

## 4.1 Yield prediction by the CPC and YP

The CPC performed a SOM training with 270 training lots. It built a SOM with 16 input neurons and 9 output neurons. The number of input nodes was chosen with regard to the process parameters, including thickness, uniformity, and critical dimension. The number of output nodes was determined to consider the analytical convenience. All the training data vectors, $\left\{\overrightarrow{x_i^{old}} = \left(x_{i1}^{old}, x_{i2}^{old}, \ldots, x_{i16}^{old}\right), 1 \le i \le 270\right\}$, were fed into the SOM in which the training algorithm adjusted the connection weights during learning. When the training finished, nine groups were obtained along with their member cases. Each group represented different patterns in process parameters.

The CPC also performed SOM classification to assort patterns of process parameters of each testing lot. The 184 testing lots have been classified, when the similarity criterion was set to 0.95. With these SOM classification results, the YP performed feature-weighted CBR for each testing lot. For example, because the testing lot 4 was classified into the case group 1, the YP performed feature-weighted CBR with the case group 1. The YP trained a BPN, having 16 input nodes, 16 hidden nodes, 2 output nodes, and 39 training cases. The 16 hidden nodes were determined by Akaike Information Criterion (AIC), which is used to determine the optimal topology of the neural network (Burnham and Anderson, 2002). The YP calculated the weight values of 16 attributes by adopting the four weighting methods. With these feature weights, the YP extracted the *k* nearest cases and generated prediction values.

## 4.2 Discussion about prediction accuracy

In order to evaluate prediction accuracy, we have compared two prediction values based on two different prediction methods:

- FW-CBR(M) - The feature-weighted CBR with the selected cases, which has been developed in this study;
- FW-CBR(A) - The feature-weighted CBR

with all cases, which was designed for performance comparison. This method uses neither SOM clustering nor SOM classification.

Both methods employed the Sensitivity/ Activity/Salience/Relevance weight values of 16 attributes. The comparison adopted varying numbers of nearest neighbors, *k*, which took odd numbers from 1 to 5. Table 2 summarizes the mean prediction accuracy of all feature-weighting methods, according to varying *k*.

The mean prediction accuracy for each testing lot was obtained by averaging prediction errors computed from ten-time experiments for each *k*. It was found that as *k* increases, the prediction errors decreased in all hybrid CBRs. The four feature-weighted CBRs with the selected cases, FW-CBR(M), outperformed the four feature-weighted CBRs with all cases, FW-CBR(A). Especially, when *k* was equal to 1, the difference in prediction accuracy was largest. The difference in the prediction accuracy between the four FW-CBR(M) methods was not small. The *Activity* feature-weighted CBR with the selected cases revealed the highest prediction accuracy.

# V. Conclusion

Yield management in the semiconductor industry has been understood as an important management practice that has to be tightly controlled. Because manufacturing process variables have non-linear complex relationship with the yield, manufacturers need an intelligent approach to pinpoint the relationship between process parameters in time.

This study devised and applied the IYPS, combining SOM and feature-weighted CBR, to prevent low yields and maintain high yields for the target semiconductor manufacturing company. As the literature review revealed, there has been few similar research to predict the yield rate of the semiconductor company while combining SOM and feature-weighted CBR. The "Activity" feature-weighting CBR with the selected case group outperformed the entire hybrid CBRs which were used for performance comparison. This hybrid CBR method also showed better performance than the existing statistical approach (a multiple regression analysis) which had been utilized by the target

<Table 2> Yield prediction results for the 184 testing lots-Mean prediction accuracy

| *k* | Sensitivity FW-CBR (M) | Activity FW-CBR (M) | Saliency FW-CBR (M) | Relevance FW-CBR (M) | Sensitivity FW-CBR (A) | Activity FW-CBR (A) | Saliency FW-CBR (A) | Relevance FW-CBR (A) |
|---|---|---|---|---|---|---|---|---|
| 1 | 0.7391 | 0.7880 | 0.7772 | 0.7337 | 0.6858 | 0.7288 | 0.7178 | 0.6838 |
| 3 | 0.7826 | 0.8207 | 0.8098 | 0.788 | 0.7219 | 0.7759 | 0.7619 | 0.7339 |
| 5 | 0.7935 | 0.8261 | 0.8152 | 0.7989 | 0.7449 | 0.7877 | 0.7906 | 0.7533 |

company.

In order to achieve higher prediction accuracy, however, the IYPS needs more process variables and data from the target company, even though the current 16 variables were determined and gathered by the manufacturing engineers under the data security enforcement. In addition, the future study needs large-scale experiments in order to do statistical comparisons. This study needs to be compared with other applicable hybrid CBR approaches in the prediction of semiconductor yield, which will be our next research topic.

## References

Ahn, H., Kim, K. and Han, I., "Global optimization of feature weights and the number of neighbors that combine in a case-based reasoning system", Expert Systems, Vol. 23, No. 5, 2006, pp. 290-301.

Burnham, K. P. and Anderson, D. R., Model selection and multimodel inference: A practical-theoretic approach, 2nd ed., Springer-Verlag, Berlin, 2002.

Caviglia, D. D., Bisio, G. M., Curatelli, F., Giovannacci, L. and Raffo, L., "Pre-placement of VLSI blocks through learning neural networks," In Proceedings of the conference on European design automation, Glasgow, Scotland, 1990, pp. 650-654.

Chien, C.-F., Wang, W.-C. and Cheng, J.-C., "Data mining for yield enhancement in semiconductor manufacturing and an empirical study", Expert Systems with Applications, Vol. 33, No. 1, 2007, pp. 192-198.

Hastie, T. and Tibshirani, R., "Discriminant adaptive nearest neighbor classification", IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 18, No. 6, 1996, pp. 607-616.

Hemani, A. and Postula, A., "Cell placement by self-organization," Neural Networks, Vol. 3, No. 4, 1990, pp. 337-338.

Hsu, S.-C. and Chien, C.-F., "Hybrid data mining approach for pattern extraction from wafer bin map to improve yield in semiconductor manufacturing", International Journal of Production Economics, Vol. 107, No. 1, 2007, pp. 88-103.

Im, K. H. and Park, S. C., "Case-based reasoning and neural network based expert system for personalization", Expert Systems with Applications, Vol. 32, 2007, pp. 77-85.

Jiang, J. W. and Jabri, M., "A new self-organization strategy for floor plan design," In Proceedings of International Joint Conference on Neural Networks, Vol. 2, Baltimore, MD, 1992, pp. 510-515.

Kang, B.-S., Lee, J. H., Shin, C. K., Yu, S. J. and Park, S. C., "Hybrid machine learning system for integrated yield management in semiconductor manufacturing", Expert Systems with Applications, Vol. 15, 1998, pp. 123-132.

Kaski, S., "Data exploration using self-organizing maps," Ph.D. dissertation, Helsinki Univ. Technol., Helsinki, 1997.

Kim, J. B., "Cooperative case-based reasoning using approximate query answering", The Journal of Information Systems (Korean Journal), Vol. 8, No. 1, 1999, pp. 27-44.

Kim, S. S. and Kyung, C. M., "Global placement of macro cells using self-organization principle," In Proceedings of IEEE International Symposium on Circuits and Systems, Vol. 5, Singapore, 1991, pp. 3122-3125.

Kiziloglu, B., Tryba, V. and Daehn, W., "Digital circuit partition by self-organizing maps: A comparison to classical methods," In Proceedings of International Joint Conference on Neural Networks, Vol. 3, Nagoya, Japan, 1993, pp. 2413-2416.

Kohonen, T., Oja, E., Simula, O., Visa, A. and Kangas, J., "Engineering applications of the self-organizing map," Proceedings of the IEEE, Vol. 84, No. 10, 1996, pp. 1358-1384.

Lee, J. H., You, S. J. and Park, S. C., "A new intelligent SOFM-based sampling plan for advanced process control", Expert Systems with Applications, Vol. 20, No. 2, 2001, pp. 133-151.

Li, T.-S. and Huang, C.-L., "Defect spatial pattern recognition using a hybrid SOM-SVM approach in semiconductor manufacturing", Expert Systems with Applications, Vol. 36, No. 1, 2009, pp. 374-385.

Liao, S.-H., "Expert system methodologies and applications-a decade review from 1995 to 2004", Expert Systems with Applications, Vol. 28, No. 1, 2005, pp. 93-103.

Liao, T. W., "An investigation of a hybrid CBR method for failure mechanisms identification", Engineering Applications of Artificial Intelligence, Vol. 17, No. 1, 2004, pp. 123-134.

Liu, H. and Motoda, H., Feature selection for Knowledge discovery and data mining, Kluwer, MA, 1998.

Marks, K. M. and Goser, K. F., "Analysis of VLSI process data based on self-organizing feature maps," In Proceedings of International Workshop on Neural Networks and Their Applications, Nanterre, France, 1988, pp. 337-348.

Roh, T., Yoo, M. and Han, I., "Integration rough set theory and case-base reasoning for the corporate credit evaluation", The Journal of Information Systems (Korean Journal), Vol. 14, No. 1, 2005, pp. 41-65.

Saridakis, K. M. and Dentsoras, A. J., "Case-DeSC: A system for case-based design with soft computing techniques", Expert Systems with Applications, Vol. 32, No. 2, 2007, pp. 641-657.

Shapiro, G. P., Brachman, R., Khabaza, T., Kloesgen, W. and Simoudis, E., "An overview of issues in developing industrial data mining and knowledge discovery applications", In Proceedings of Second International Conference on Knowledge Discovery and Data mining, 1996, pp. 89-95.

Shin, C. K. and Park, S. C., "Memory and neural network based expert system", Expert Systems with Applications, Vol. 16, 1999, pp. 145-155.

Shin, C. K., Yun, U. T., Kim, H. K. and Park, S. C., "A hybrid approach of neural network and memory-based learning to data mining", IEEE Transactions on Neural Networks, Vol. 11, No. 3, 2000, pp. 637-646.

Tan, K. H., Lim, C. P., Platts, K. and Koay, H. S., "An intelligent decision support system for manufacturing technology investments", International Journal of Production Economics, Vol. 104, No. 1, 2006, pp. 179-190.

Tsay, M.-Y., "Journal self-citation study for semiconductor literature: Synchronous and diachronous approach", Information Processing & Management, Vol. 42, No. 6, 2006, pp. 1567-1577.

Wang, C.-H., "Recognition of semiconductor defect patterns using spatial filtering and spectral clustering", Expert Systems with Applications, Vol. 34, No. 3, 2008, pp. 1914-1923.

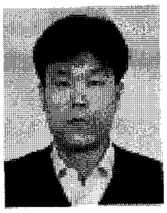
Wettschereck, D. and Aha, D. W., "Weighting features", In Proceedings of the First International Conference on Case-Based Reasoning, A. Aamodt and M. Veloso (eds), pp. 347-358, 1995.

Wettschereck, D., Aha, D. W. and Mohri, T., "A review and empirical evaluation of feature weighting methods for a class of lazy learning algorithms", Artificial Intelligence Review, Vol. 11, 1997, pp. 273-314.

Yang, B.-S., Han, T. and Kim, Y.-S., "Integration of ART-Kohonen neural network and case-based reasoning for intelligent fault diagnosis", Expert Systems with Applications, Vol. 26, No. 3, 2004, pp. 387-395.

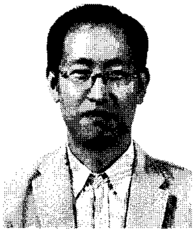
Zhang, L., Coenen, F. and Leng, P., "Formalising optimal feature weight setting in case based diagnosis as linear programming problems", Knowledge-Based Systems, Vol. 15, No. 7, 2002, pp. 391-398.

**이장희 (Jang Hee Lee)**



고려대학교 수학교육 이학사, KAIST 산업공학 공학석사, KAIST에서 산업공학 공학박사 학위를 취득하였다. 삼성전자 반도체를 거쳐 현재 한국기술교육대학교 산업경영학부 교수로 재직 중이다. 주요 관심분야는 데이터 마이닝, 지능형 정보시스템, CRM/SCM, 6시그마 품질경영 등이다.

**하성호 (Sung Ho Ha)**



연세대학교 경영학과를 졸업하고, 한국과학기술원에서 정보시스템으로 공학석사, 공학박사를 받았다. 현재 경북대학교 경영학부 부교수로 재직 중이다. 다수 국내외 저널의 편집위원으로 활동하고 있으며 관심분야는 지능형정보시스템, 데이터마이닝, e-비즈니스, 지식서비스 등이 있다.

<Abstract>

# 소프트컴퓨팅 기법을 활용하는 지능적인 반도체 수율 분류 시스템

이장희 · 하성호

생산 수율은 비선형관계를 지닌 여러 요인들에 의해 영향을 받기 때문에 반도체 생산의 경우 예측이 어렵다. 본 논문에서 저자들은 사례기반추론과 자기조직화신경망 기반의 데이터마이닝 기법을 활용하여 수율의 높고 낮음을 밝히는 지능화된 수율예측시스템을 제시한다. 이 시스템은 자기조직화신경망을 사용하여 생산 로트의 공정파라미터 패턴을 파악하고 속성가중치 기반의 사례기반추론을 통해 신규 로트의 수율 수준을 예측한다. 이때 속성가중치는 역전파인공신경망을 통해 계산된다. 웹기반 시스템이 개발되고, 반도체 생산 기업의 실제 자료를 적용하여 본 시스템의 효율을 검증하고 평가한다.

***Keywords***: 수율예측, 자기조직화맵, 가중속성, 사례기반추론

* 이 논문은 2009년 11월 9일 접수하여 1차 수정을 거쳐 2010년 2월 4일 게재 확정되었습니다.