Research Article

Precise and Accurate Job Cycle Time Forecasting in a Wafer Fabrication Factory with a Fuzzy Data Mining Approach

Toly Chen and Richard Romanowski

Department of Industrial Engineering and Systems Management, Feng Chia University, 100 Wenhwa Road, Seatwen, Taichung 407, Taiwan

Correspondence should be addressed to Toly Chen; tcchen@fcu.edu.tw

Received 5 November 2012; Revised 18 February 2013; Accepted 12 March 2013

Academic Editor: Sabrina Senatore

Copyright © 2013 T. Chen and R. Romanowski. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Many data mining methods have been proposed to improve the precision and accuracy of job cycle time forecasts for wafer fabrication factories. This study presents a fuzzy data mining approach based on an innovative fuzzy backpropagation network (FBPN) that determines the lower and upper bounds of the job cycle time. Forecasting accuracy is also significantly improved by a combination of principal component analysis (PCA), fuzzy c-means (FCM), and FBPN. An applied case that uses data collected from a wafer fabrication factory illustrates this fuzzy data mining approach. For this applied case, the proposed methodology performs better than six existing data mining approaches.

1. Introduction

To forecast the job cycle time in a wafer fabrication factory is difficult because each wafer fabrication factory is a complicated production system with idiosyncratic features such as changing demand, a variety of product types and priorities, equipment unreliability, unbalanced capacity, job reentry into machines, alternative machines, sequence-dependent setup times, and shifting bottlenecks. An average job cycle time is several months with hundreds of hours of standard deviation. Many studies have shown that accurately predicting the cycle/completion times for such large systems is very difficult [2–4].

There are two approaches to forecasting the cycle time of a job. The first approach, the input-output relationship approach, is to determine certain factors (e.g., average waiting time, queuing length, utilization, future release plans, bottlenecks, etc.) that influence the job cycle time, and then to apply different approaches (e.g., multiple linear regression (MLR), artificial neural networks (ANN), etc.) to model the relationship between the job cycle time and these factors, in order to forecast the cycle time of a new job. The second approach, the time-series approach, is to treat fluctuations in the job cycle time as a type of time series. Many approaches, for example, moving average (MA), weighted moving average (WMA),

exponential smoothing (ES), MLR, ANN, autoregressive integrated moving average (ARIMA), and others, can be applied to forecast the cycle time of a new job. The cycle time of a job is usually predicted when the job is released into the factory, but several months are required for the job to complete all operations. In such a case, the production controller may need to forecast the cycle time of job number $k+10000$ according to the historical data of jobs $1 \sim k$. This is a difficult problem for time-series methods; the present study contributes a useful approach to this problem.

Existing approaches to job cycle time estimation for a wafer fabrication factory can be classified into six categories: statistical analysis, production simulation (PS), ANN, case-based reasoning (CBR), fuzzy modeling methods, and hybrid approaches [5]. Many approaches are in fact special types of data mining methods. Table 1 summarizes references related to these categories. Data mining methods that have been extensively applied to job cycle time forecasting include clustering, genetic algorithms (GA), ANNs, decision trees, and hybrid approaches. The relevant references are reviewed below.

Backus et al. [1] compared the performances of clustering, K -nearest neighbors (i.e., CBR), classification and regression tree (CART), and ANN; CART gave the best performance

TABLE 1: References relevant to the six categories.

Category	Related references
Statistical analysis	Pearn et al. [2], Chien et al. [4], and Tai et al. [16]
PS	Sivakumar and Chong [17], Okumura [18], and Veeger et al. [19]
ANN	Chen [5], Chang and Hsieh [9], Sha and Hsu [10], Chang et al. [8], Backus et al. [1], Chien et al. [4], Meidan et al. [6], and Tirkel [15]
CBR	Chang et al. [20], Chiu et al. [21], and Chang et al. [22]
Fuzzy modeling	FBPN [5], Chang et al. [8], kM-FBPN [11], look-ahead SOM-FBPN [3, 23, 24], post-classifying FBPN-BPN [13, 14], and bi-directional classifying FCM-FBPN-RBF [25]
Hybrid methods	SOM-WM [26], SOM-BPN [27], kM-FBPN [11], FCM-FBPN [28, 29], look-ahead SOM-FBPN [3, 23, 24], post-classifying FBPN-BPN [13, 14], and bi-directional classifying FCM-FBPN-RBF [25]

of that experiment. Meidan et al. [6] focused on the choice of key factors in cycle time forecasting. They compared multiple linear regression (i.e., statistical analysis), ANN, and a selective naive Bayesian classifier (SNBC); they found that their ANN had the best performance.

Wang and Mendel [7] proposed a fuzzy modeling method called the WM method; the first step of that method was partitioning the range of each input variable into several fuzzy intervals. Chang et al. [8] modified the first step of the WM method with a simple GA and proposed an evolving fuzzy rule (EFR) approach to predict the cycle time of a job in a wafer fabrication factory. Their EFR approach outperformed CBR and ANN in terms of forecasting accuracy.

A BPN is an effective tool for modeling complex physical systems described by sets of different equations. BPNs are useful for prediction, control, and design purposes; they can offer both effectiveness (high forecasting accuracy) and efficiency (short execution time). Chang and Hsieh [9], Chang et al. [8], and Sha and Hsu [10] all predicted the cycle times of jobs in wafer fabrication factories with backpropagation networks (BPNs) that each had a single hidden layer. These BPNs delivered better average forecasting accuracy than statistical analysis approaches, as measured by root mean squared error (RMSE). For example, an improvement of about 40% in RMSE was achieved in Chang et al. [8]. Chen [5] constructed a fuzzy backpropagation network (FBPN) that incorporated expert opinions to modify the inputs of the FBPN. Chen's FBPN surpassed a crisp BPN especially with respect to efficiency. Chen [11] incorporated the job release plan of a wafer fabrication factory into a BPN, and constructed a "look-ahead" BPN for the same purpose, which led to an average reduction of 12% in RMSE. Beeg [12] and Chen et al. [13, 14] predicted the cycle time of a job in a ramping-up wafer fabrication factory. Chen et al. used a BPN-based method; Beeg tried to figure out the impact of utilization for the cycle time. Tirkel [15] applied two data mining approaches—decision trees and ANN—and concluded that ANN was more accurate.

Chang and Liao [26] combined a self-organizing map and the Wang and Mendel method (SOM-WM); a job was classified using a SOM before the job cycle time was predicted by the WM method. Chen [11] constructed a look-ahead k-means-fuzzy backpropagation network (kM-FBPN) for the same purpose and discussed the effects of using different look-ahead functions. More recently, Chen et al. [23] proposed a look-ahead SOM-FBPN approach for job cycle time forecasting in a semiconductor factory; a set of fuzzy inference rules was developed to evaluate the achievability of a cycle time forecast. Subsequently, Chen et al. [3] added a selective time allowance to the look-ahead SOM-FBPN approach to determine internal due dates. Further, Chen [24] showed that a combined SOM and FBPN could be improved by feeding back the forecasting error by the FBPN to adjust the classification results of the SOM. Chen et al. [13, 14] proposed a postclassification FBPN-BPN approach in which a job was not preclassified; the job was classified after the cycle time had been predicted. Experimental results showed that the post-classification approach was better than some preclassification approaches in some cases. In order to combine the advantages of preclassifying and post-classifying approaches, Chen [25] proposed a bidirectional classifying approach that combined FCM, FBPN, and a radial basis function network (RBF), in which jobs are not only preclassified but also post-classified. Instead of bi-directional classification, Chien et al. [4] used nonlinear regression equations and then related the forecasting error to some factory conditions and job attributes with a backpropagation network (BPN) to improve the forecasting accuracy.

Table 2 summarizes the data mining methods in this field. The advantages and disadvantages of these methods are compared in Table 3.

To improve the precision and accuracy of job cycle time forecasting in a wafer fabrication factory, this study presents a fuzzy data mining approach that emphasizes both accuracy and precision.

- (1) Accuracy: the forecast value should be as close as possible to the actual value.
- (2) Precision: a narrow interval containing the actual value should be established.

Chen [30] defined several methods to measure the precision and accuracy of a fuzzy forecasting method and noted that precision has seldom been considered. However, past studies have confirmed that precision and accuracy are closely related. Song and Chissom [31] proposed a first-order time-variant model to forecast enrollment and considered the effect of the model basis on the forecasting precision. Specifically, if the space of feasible solutions is narrowed, it becomes more likely that the actual value can be found [30, 32].

The fuzzy data mining approach has the following innovative characteristics

- (1) Some factors used to estimate the cycle time are dependent on each other, which may cause problems in classifying jobs and in fitting the relationship between the job cycle time and these factors. To solve this problem, PCA, FCM, and FBPN are combined to

TABLE 2: Data mining methods for job cycle time forecasting.

Category	Examples
Clustering	Backus et al. [1], Meidan et al. [6]
GA	Chang et al. [8]
Decision trees	CART [1], SNBC [6], and Tirkel [15]
ANN	Chen [5], Chang and Hsieh [9], Sha and Hsu [10], Chang et al. [8], Backus et al. [1], Chien et al. [4], Meidan et al. [6], and Tirkel [15]
Hybrid approaches	SOM-WM [26], kM-FBPN [11, 29], look-ahead SOM-FBPN [3, 23, 24], post-classifying FBPN-BPN [13, 14], and bi-directional classifying FCM-FBPN-RBF [25]

predict the job cycle time, which is an innovation in this field.

- (2) The job cycle time is predicted by an effective FBPN approach. For the job cycle time, there are upper and lower bounds that constrain the forecast to be within a possible range. The FBPN approach from Chen and Wang's FBPN research [33] has been modified for this purpose. In Chen and Wang's FBPN approach two nonlinear programming (NP) models are solved to determine the upper and lower bounds of the job cycle time. However, the NP models involve complicated constraints and therefore are difficult to solve. In addition, the NP models will become too huge if many jobs are to be considered. To solve these problems, in this study, the upper and lower bounds are determined in a simpler way.

The differences between the proposed methodology and various previous methods are summarized in Table 4.

The remainder of this paper is organized as follows. Section 2 introduces the proposed methodology, which is composed of six steps. Section 3 demonstrates the proposed method with an illustrative example, then presents another case with data collected from a real wafer fabrication factory, and makes comparisons with some existing approaches. Finally, concluding remarks and directions for future research are given in Section 4.

2. Methodology

The fuzzy data mining procedure consists of several steps.

Step 1. Replacing parameters using PCA.

Step 2. Classifying jobs using fuzzy c-means (FCM).

Step 3. Forecasting the cycle times of jobs in each category using a FBPN.

Step 4. Updating the upper and lower bounds.

Step 5. Evaluating and comparing the forecasting precision and accuracy of the proposed methodology.

FIGURE 1: The flowchart of the proposed methodology.

Step 6. If the improvement in the forecasting precision or accuracy becomes negligible, stop; otherwise, return to Step 3.

A flow chart of the proposed methodology is shown in Figure 1. These steps can be compared to the basic steps of data mining, as shown in Table 5.

2.1. Step 1: Variable Replacement Using PCA. First, PCA is used to replace the inputs to the FBPN. A series of linear combinations of the original variables forms a new variable so that these new variables are as unrelated to each other as possible. Although there are more advanced applications of PCA, in this study PCA is used to enhance the efficiency of FBPN training. PCA consists of the four following steps.

- (1) Raw data standardization: to eliminate dimensional differences and large numerical differences the original variables are standardized [13, 14]:

$$\begin{aligned}
 x_{ji}^* &= \frac{x_{ji} - \bar{x}_i}{\sigma_i}, \\
 \bar{x}_i &= \frac{\sum_{j=1}^n x_{ji}}{n}, \\
 \sigma_i &= \sqrt{\frac{\sum_{j=1}^n (x_{ji} - \bar{x}_i)^2}{n-1}},
 \end{aligned} \tag{1}$$

TABLE 3: Advantages and disadvantages of various data mining approaches for job cycle time forecasting.

Approach	Advantages	Disadvantages
Clustering	(i) Easiest to use (ii) Can be used with other methods	(i) Not accurate (ii) No method to determine the optimal number of clusters
GA	(i) Accurate (ii) Can be used with other methods	Time-consuming iterations
Decision trees	(i) Handles missing values well (ii) Robust to outliers	High variance of tree structure
ANN	(i) Accurate (ii) Robust to outliers	(i) Difficult to use (ii) Time-consuming training (iii) Possible overfitting
Hybrid approaches	Very accurate	(i) Very difficult to use (ii) Very time consuming

TABLE 4: The differences between the proposed methodology and the previous methods.

Method	Variable replacement	Job pre-classification	Job post-classification	Determining the bounds
SOM-WM, kM-FBPN, Look-ahead SOM-FBPN	No	Yes	No	Not discussed
Post-classifying FBPN-BPN	No	No	Yes	Not discussed
Bi-directional classifying FCM-FBPN-RBF	No	Yes	Yes	Not discussed
BPN-NP	No	No	No	Difficult
The proposed methodology	Yes	Yes	Yes	Easy

TABLE 5: The proposed methodology compared to data mining.

Data mining steps	The proposed methodology
Application domain identification	Job cycle time forecasting for a wafer fabrication factory
Target dataset selection	The most recent PROMIS data of a major product type with normal priority
Data preprocessing	PCA, partial normalization (i) FCM for job classification (ii) FBPN to model the relationship between the job cycle time and six job parameters
Data mining	
Knowledge extraction	Training data (3/4 of the collected data)
Knowledge application	Testing data (the remaining data)
Knowledge evaluation	Forecasting precision (the average range) and accuracy (MAE, MAPE, RMSE)

*PROMIS: production management information system; MAE: mean absolute error; MAPE: mean absolute percentage error.

where x_{ji} is the i th attribute of job j , $j = 1 \sim n$; $\bar{x}_i$ and σ_i indicate the mean and standard deviation of variable i , respectively.

- (2) Establishment of the correlation matrix R is as follows:

$$R = \frac{1}{n-1} X^{*T} X^*, \quad (2)$$

where X^* is the standardized data matrix. The eigenvalues and eigenvectors of R are calculated and represented as $\lambda_1 \sim \lambda_m$ and $u_1 \sim u_m$, respectively; $\lambda_1 \geq \lambda_2 \geq \dots \geq \lambda_m$.

- (3) Determination of the number of principal components: the variance contribution rate is calculated as

$$\eta_q = \frac{\lambda_q}{\sum_{r=1}^m \lambda_r} \cdot 100\%, \quad (3)$$

and the accumulated variance contribution rate is

$$\eta_\Sigma(p) = \sum_{q=1}^p \eta_q. \quad (4)$$

Choose the smallest P value such that $\eta_\Sigma(p)$ ranges from 85% to 90%. A Pareto analysis chart can be used to show the percentage of variability that can be attributed to each principal component.

- (4) Formation of the following matrices:

$$U_{m \times p} = [u_1, u_2, \dots, u_p], \quad (5)$$

$$Z_{n \times p} = X_{n \times m}^* U_{m \times p}.$$

$Z_{n \times p} = [z_{jq}]$ ($j = 1 \sim n$; $q = 1 \sim p$) are the component scores. These scores contain the coordinates of the original data in the new coordinate system defined by the principal components. They will be used as the new inputs to the FBPN.

2.2. Step 2: Classifying Jobs Using FCM. After PCA, examples are classified by FCM. If a crisp clustering method is applied instead, then it is possible that some clusters will have very few examples. FCM avoids the problems of crisp clustering, because in FCM, an example belongs to multiple clusters to

different degrees. Similarly, in probability theory the naïve Bayes method provides a probability that each item belongs to a class. However, the classification of jobs in FCM can include subjective judgments.

FCM classifies jobs by minimizing the following objective function:

$$\text{Min} \sum_{k=1}^K \sum_{j=1}^n \mu_{j(k)}^m e_{j(k)}^2, \quad (6)$$

where K is the required number of categories; n is the number of jobs; $\mu_{j(k)}$ indicates the degree of membership that job j has in category k ; $e_{j(k)}$ measures the distance from job j to the centroid of category k ; $m \in [1, \infty)$ is a parameter to adjust the fuzziness and is usually set to 2. The procedure of FCM is as follows.

- (1) Normalize the input data.
- (2) Produce a preliminary clustering result.
- (3) Iterations calculate the centroid of each category as

$$\begin{aligned} \bar{z}_{(k)} &= \{\bar{z}_{(k)q}\}, \quad k = 1 \sim K, \\ \bar{z}_{(k)q} &= \frac{\sum_{j=1}^n \mu_{j(k)}^m z_{jq}}{\sum_{j=1}^n \mu_{j(k)}^m}, \quad k = 1 \sim K, \quad q = 1 \sim p, \\ \mu_{j(k)} &= \frac{1}{\sum_{g=1}^K (e_{j(k)}/e_{j(g)})^{2/(m-1)}}, \quad j = 1 \sim n, \quad k = 1 \sim K, \\ e_{j(k)} &= \sqrt{\sum_{q=1}^p (z_{jq} - \bar{z}_{(k)q})^2}, \quad j = 1 \sim n, \quad k = 1 \sim K, \end{aligned} \quad (7)$$

where $\bar{z}_{(k)}$ is the centroid of category k . $\mu_{j(k)}^{(t)}$ is the degree of membership that job j has in category k after the t th iteration.

- (4) Remeasure the distance from each job to the centroid of each category and then recalculate the corresponding membership.
- (5) Stop if the following condition is met. Otherwise, return to Step (3):

$$\max_k \max_j |\mu_{j(k)}^{(t)} - \mu_{j(k)}^{(t-1)}| < d, \quad (8)$$

where d is a real number representing the membership convergence threshold.

The performance of FCM is affected by the settings of the initial values; initialization can be repeated multiple times in order to find the optimal solution. Finally, the separate

distance test (S test) proposed by Xie and Beni [34] can be applied to determine the optimal number of categories K :

$$\begin{aligned} \text{Min} \quad & S \\ \text{subject to} \quad & J_m = \sum_{k=1}^K \sum_{j=1}^n \mu_{j(k)}^m e_{j(k)}^2, \\ & e_{\min}^2 = \min_{k1 \neq k2} \left(\sum_{q=1}^p (\bar{z}_{(k1)q} - \bar{z}_{(k2)q})^2 \right), \quad (9) \\ & S = \frac{J_m}{n \times e_{\min}^2}, \quad K \in Z^+. \end{aligned}$$

The K value that minimizes S determines the optimal number of categories.

2.3. Step 3: Forecasting the Cycle Times of Jobs in Each Category Using a FBPN. All categories in the present study are considered by the FBPN. There are different types of FBPNs, for example, Nomura et al. [35], Chen [5], Lin [36], and Chen and Wang [33]. Among them, Chen and Wang's FBPN is unique since it is aimed at optimizing precision. For this reason, it is chosen for this study. This FBPN is a proven technology that has incorporated many features of prior artificial neural networks, which have solved a wide variety of problems characterized by sets of different equations. Some special problems can benefit from specialized tools, such as compositional pattern-producing networks, cascading neural networks, and dynamic neural networks, but for the present job cycle time problem, a well-trained FBPN with an optimized structure can still produce very good results. All parameters in the FBPN are expressed in triangular fuzzy numbers (TFNs). There are various types of fuzzy numbers with different shapes; among them, the triangular type is easily implemented and has been universally applied to numerous applications (e.g., [37, 38]). Although the proposed methodology uses TFNs, it can be easily modified to use other types of fuzzy numbers, such as trapezoidal fuzzy numbers or bounded LR-type fuzzy numbers.

The FBPN is configured as follows.

- (1) Inputs: there are j sets of inputs; each set consists of the new factors determined by PCA from the j th job. These factors have to be partially normalized so that their values fall within [0.1, 0.9] [13, 14].
- (2) Single hidden layer: generally one hidden layer gives the best convergence results for the FBPN.
- (3) For simplicity, the number of neurons in the hidden layer is twice the number in the input layer. An increase in the number of hidden-layer nodes lessens the output errors for the training examples, but increases the errors for novel examples. This phenomenon is often called "overfitting." Some research has considered the connections between the complexity of a FBPN, the performance for the training data, and the number of examples; noteworthy research includes Akaike's information criterion (AIC) [39]

and Rissanen's minimum description length (MDL) [40].

- (4) Output: the (normalized) cycle time forecast of the example.

The procedure for determining the parameter values is now described. After pre-classification, a portion of the adopted examples in each category is fed as "training examples" into the FBPN to determine the parameter values. Two phases are involved at the training stage. First, in the forward phase, inputs are multiplied with weights, summed, and transferred to the hidden layer. Then, activated signals are outputted from the hidden layer as

$$\begin{aligned}\tilde{h}_l &= (h_{l1}, h_{l2}, h_{l3}) \\ &= \frac{1}{1 + e^{-\tilde{n}_l^h}} \\ &= \left(\frac{1}{1 + e^{-n_{l1}^h}}, \frac{1}{1 + e^{-n_{l2}^h}}, \frac{1}{1 + e^{-n_{l3}^h}} \right),\end{aligned}\quad (10)$$

where

$$\begin{aligned}\tilde{n}_l^h &= (n_{l1}^h, n_{l2}^h, n_{l3}^h) \\ &= \tilde{I}_l^h (-) \tilde{\theta}_l^h \\ &= (I_{l1}^h - \theta_{l3}^h, I_{l2}^h - \theta_{l2}^h, I_{l3}^h - \theta_{l1}^h), \\ \tilde{I}_l^h &= (I_{l1}^h, I_{l2}^h, I_{l3}^h) \\ &= \sum_{\text{all } k} \tilde{w}_{kl}^h \cdot x_k \\ &= \left(\sum_{\text{all } k} \min(w_{kl1}^h x_k, w_{kl3}^h x_k), \right. \\ &\quad \left. \sum_{\text{all } k} w_{xk2}^h x_k, \sum_{\text{all } k} \max(w_{kl1}^h x_k, w_{kl3}^h x_k) \right)\end{aligned}\quad (11)$$

$\tilde{h}_l$ is the output from hidden-layer node l , $l = 1 \sim L$; $\tilde{\theta}_l^h$ is the threshold for screening out weak signals by hidden-layer node l ; $\tilde{w}_{kl}^h$ is the weight of the connection between input node k and hidden-layer node l , $k = 1 \sim p$; $l = 1 \sim L$; x_{ij} is the k th input, $k = 1 \sim p$. The remaining parameters are transition variables. $(-)$ and $(\times)$ denote fuzzy subtraction and multiplication, respectively.

$\tilde{I}_l^h$ s are also transferred to the output layer with the same procedure. Finally, the output of the FBPN is generated as follows:

$$\begin{aligned}\tilde{o}_j &= (o_{j1}, o_{j2}, o_{j3}) \\ &= \frac{1}{1 + e^{-\tilde{n}^o}} \\ &= \left(\frac{1}{1 + e^{-n_1^o}}, \frac{1}{1 + e^{-n_2^o}}, \frac{1}{1 + e^{-n_3^o}} \right),\end{aligned}\quad (12)$$

where

$$\tilde{n}^o = (n_1^o, n_2^o, n_3^o) = \tilde{I}^o (-) \tilde{\theta}^o = (I_1^o - \theta_3^o, I_2^o - \theta_2^o, I_3^o - \theta_1^o), \quad (13)$$

$$\begin{aligned}\tilde{I}^o &= (I_1^o, I_2^o, I_3^o) \\ &= \sum_{\text{all } l} \tilde{w}_l^o (\times) \tilde{h}_l \\ &\cong \left(\sum_{\text{all } l} \min(w_{l1}^o h_{l1}, w_{l3}^o h_{l3}), \right. \\ &\quad \left. \sum_{\text{all } l} w_{l2}^o h_{l2}, \sum_{\text{all } l} \max(w_{l1}^o h_{l1}, w_{l3}^o h_{l3}) \right).\end{aligned}\quad (14)$$

$\tilde{o}_j$ is the network output, which is the normalized value of the cycle time forecast of job j ; $\tilde{\theta}^o$ is the threshold for screening out weak signals by the output node; $\tilde{w}_l^o$ is the weight of the connection between hidden-layer node l and the output node; $l = 1 \sim L$.

Subsequently, in the backward phase, the training of the FBPN is decomposed into three subtasks: determining the center value and upper and lower bounds of the parameters.

First, to determine the center of each parameter (such as w_{kl2}^h , θ_{l2}^h , w_{l2}^o , and θ_2^o), the FBPN is treated as a crisp network. Algorithms applicable for this purpose include the gradient descent algorithms, the conjugate gradient algorithms, the Levenberg-Marquardt algorithm, and others. In this study, the Levenberg-Marquardt algorithm is applied. The Levenberg-Marquardt algorithm was designed for training with second-order speed without having to compute the Hessian matrix. It uses approximation and updates the network parameters in a Newton-like way, as described below.

The network parameters are placed in vector $\beta = [w_{11}^h, \dots, w_{pL}^h, \theta_1^h, \dots, \theta_L^h, w_1^o, \dots, w_L^o, \theta^o]$. The network output o_j can be represented with $f(\mathbf{x}_j, \beta)$. The objective function of the FBPN is to minimize RMSE or equivalently the sum of squared error (SSE):

$$\text{SSE}(\beta) = \sum_{j=1}^n (N(\text{CT}_j) - f(\mathbf{x}_j, \beta))^2. \quad (15)$$

The Levenberg-Marquardt algorithm is an iterative procedure. In the beginning, the user should specify the initial values of the network parameter vector β . It is a common practice to set $\beta^T = (1, 1, \dots, 1)$. In each step, the parameter vector β is replaced by a new estimate $\beta + \delta$, where $\delta = [\Delta w_{11}^h, \dots, \Delta w_{pL}^h, \Delta \theta_1^h, \dots, \Delta \theta_L^h, \Delta w_1^o, \dots, \Delta w_L^o, \Delta \theta^o]$. The network output becomes $f(\mathbf{x}_j, \beta + \delta)$; it is approximated by its linearization as

$$f(\mathbf{x}_j, \beta + \delta) \approx f(\mathbf{x}_j, \beta) + \mathbf{J}_j \delta, \quad (16)$$

where

$$\mathbf{J}_j = \frac{\partial f(\mathbf{x}_j, \beta)}{\partial \beta} \quad (17)$$

is the gradient vector of f with respect to β . Substituting (16) into (15) gives

$$\text{SSE}(\beta + \delta) \approx \sum_{j=1}^n (N(CT_j) - f(x_j, \beta) - J_j \delta)^2. \quad (18)$$

When the network reaches the optimal solution, the gradient of SSE with respect to δ will be zero. Taking the derivative of $\text{SSE}(\beta + \delta)$ with respect to δ and setting the result to zero gives

$$(J^T J) \delta = J^T (N(CT_j) - f(x_j, \beta)), \quad (19)$$

where J is the Jacobian matrix containing the first derivative of network error with respect to the weights and biases. Equation (19) includes a set of linear equations that can be solved for δ .

2.4. Step 4: Determining the Upper Bounds of Parameters in the FBPN. In Chen and Wang [33] and Chen and Lin [41], an NP model is constructed to adjust the connection weights and thresholds in the FBPN; such an NP model is not easy to solve. The proposed methodology only makes iterative adjustments to the threshold on the output node. This way is much simpler and can also achieve good results.

Substituting (13) into (12) gives

$$o_{j2} = \frac{1}{1 + e^{-n_{j2}^o}} = \frac{1}{1 + e^{-(I_{j2}^o - \theta_2^o)}} = \frac{1}{1 + e^{\theta_2^o - I_{j2}^o}}. \quad (20)$$

Therefore,

$$\ln\left(\frac{1}{o_{j2}} - 1\right) = \theta_2^o - I_{j2}^o. \quad (21)$$

So

$$I_{j2}^o = \theta_2^o - \ln\left(\frac{1}{o_{j2}} - 1\right). \quad (22)$$

Assume that the adjustment made to the output node threshold is denoted as $\Delta\theta^o = \theta_3^o - \theta_2^o$. After adjustment, the output from the new FBPN, o_{j3} , determines the upper bound of the cycle time:

$$o_{j3} = \frac{1}{1 + e^{-n_{j3}^o}}, \quad (23)$$

where

$$n_{j3}^o = I_{j3}^o - \theta_3^o = I_{j3}^o - (\theta_2^o + \Delta\theta^o). \quad (24)$$

Substituting (24) into (23) gives

$$o_{j3} = \frac{1}{1 + e^{-(I_{j3}^o - \theta_2^o - \Delta\theta^o)}}, \quad (25)$$

and substituting (22) into (25) gives

$$\begin{aligned} o_{j3} &= \frac{1}{1 + e^{-(\theta_2^o - \ln(1/o_{j2} - 1) - \theta_2^o - \Delta\theta^o)}} \\ &= \frac{1}{1 + e^{\ln(1/o_{j2} - 1) + \Delta\theta^o}} \\ &= \frac{1}{1 + e^{\Delta\theta^o} (1/o_{j2} - 1)}. \end{aligned} \quad (26)$$

FIGURE 2: Iterative reduction of the upper bound.

Obviously, the maximum of $\Delta\theta^o$ determines the lowest upper bound, since o_{j3} is the upper bound of the cycle time, $o_{j3} \geq N(CT_j)$,

$$\frac{1}{1 + e^{\ln(1/o_{j2} - 1) + \Delta\theta^o}} \geq N(CT_j), \quad (27)$$

$$\Delta\theta^o \leq \ln\left(\frac{1}{N(CT_j)} - 1\right) - \ln\left(\frac{1}{o_{j2}} - 1\right). \quad (28)$$

Equation (28) holds for all jobs, so

$$\Delta\theta^o \leq \min_j \left(\ln\left(\frac{1}{N(CT_j)} - 1\right) - \ln\left(\frac{1}{o_{j2}} - 1\right) \right). \quad (29)$$

According to (26), the optimal value of $\Delta\theta^o$ should be set to the maximum possible value:

$$\Delta\theta^{o*} = \min_j \left(\ln\left(\frac{1}{N(CT_j)} - 1\right) - \ln\left(\frac{1}{o_{j2}} - 1\right) \right). \quad (30)$$

Then the FBPN's optimization results depend on the initial conditions and therefore are different at every iteration. Assume that the optimal value of o_{j3} in the t th iteration is denoted by $o_{j3}(t)$, then, after some iterations,

$$o_{j3}(\text{all iterations}) = \min_t o_{j3}(t). \quad (31)$$

In this way, the upper bound of the cycle time is decreased gradually (see Figure 2). Another merit of this approach is that it does not rely on the parameters of the FBPN.

Theorem 1. The upper bound can be found by considering only jobs with $N(CT_j) \geq o_{j2}$.

Proof. Divide jobs into two categories C_1 (jobs with $N(CT_j) < o_{j2}$) and C_2 (jobs with $N(CT_j) \geq o_{j2}$). Equation (30) becomes

$$\begin{aligned} \Delta\theta^{o*} &= \min \left(\min_{j \in C_1} \left(\ln \left(\frac{1}{N(CT_j)} - 1 \right) - \ln \left(\frac{1}{o_{j2}} - 1 \right) \right), \right. \\ &\quad \left. \min_{j \in C_2} \left(\ln \left(\frac{1}{N(CT_j)} - 1 \right) - \ln \left(\frac{1}{o_{j2}} - 1 \right) \right) \right). \end{aligned} \quad (32)$$

If $N(CT_j) < o_{j2}$, then

$$\ln \left(\frac{1}{N(CT_j)} - 1 \right) - \ln \left(\frac{1}{o_{j2}} - 1 \right) > 0. \quad (33)$$

Therefore,

$$\min_{j \in C_1} \left(\ln \left(\frac{1}{N(CT_j)} - 1 \right) - \ln \left(\frac{1}{o_{j2}} - 1 \right) \right) > 0. \quad (34)$$

In contrast, if $N(CT_j) \geq o_{j2}$, then

$$\begin{aligned} \ln \left(\frac{1}{N(CT_j)} - 1 \right) - \ln \left(\frac{1}{o_{j2}} - 1 \right) &\leq 0, \\ \min_{j \in C_2} \left(\ln \left(\frac{1}{N(CT_j)} - 1 \right) - \ln \left(\frac{1}{o_{j2}} - 1 \right) \right) &\leq 0. \end{aligned} \quad (35)$$

Finally,

$$\Delta\theta^{o*} = \min_{j \in C_2} \left(\ln \left(\frac{1}{N(CT_j)} - 1 \right) - \ln \left(\frac{1}{o_{j2}} - 1 \right) \right). \quad (36)$$

Theorem 1 is proved. $\square$

2.5. Step 4: Determining the Lower Bounds of Parameters in the FBP. In a similar way, the threshold on the output node can be modified to determine the lower bound of the cycle time forecast so that the actual value will be greater than the lower bound of the network output. The optimal value of $\Delta\theta^o$ can be obtained as

$$\Delta\theta^{o*} = \max_j \left(\ln \left(\frac{1}{N(CT_j)} - 1 \right) - \ln \left(\frac{1}{o_{j2}} - 1 \right) \right). \quad (37)$$

Theorem 2. *The lower bound can be found by considering only jobs with $N(CT_j) < o_{j2}$.*

The proof of Theorem 2 resembles that of Theorem 1.

Proof. Assume that the optimal value of o_{j1} in the t th replication is indicated with $o_{j1}(t)$, then after some iterations,

$$o_{j1}(\text{all iterations}) = \max_t o_{j1}(t). \quad (38)$$

In this way, the upper bound of the cycle time is increased gradually. $\Delta\theta^{o*}$ does not rely on the parameters of the FBP. $\square$

3. Application and Analyses

To demonstrate the application of the proposed methodology, an illustrative example containing the data of 40 jobs (see Table 6) was used.

We standardize the data and obtain the correlation matrix R . The eigenvalues and eigenvectors of R are calculated as

$$\begin{aligned} \lambda_1 &= 2.66, & \lambda_2 &= 1.15, & \lambda_3 &= 0.94, \\ \lambda_4 &= 0.83, & \lambda_5 &= 0.25, & \lambda_6 &= 0.02, \end{aligned} \quad (39)$$

$$u_1 = \begin{bmatrix} 0.20 \\ 0.52 \\ 0.16 \\ 0.56 \\ 0.07 \\ 0.59 \end{bmatrix}, \quad u_2 = \begin{bmatrix} 0.47 \\ -0.19 \\ 0.68 \\ -0.19 \\ -0.49 \\ 0.05 \end{bmatrix},$$

$$u_3 = \begin{bmatrix} -0.27 \\ 0.31 \\ -0.32 \\ 0.03 \\ -0.85 \\ -0.02 \end{bmatrix}, \quad u_4 = \begin{bmatrix} 0.81 \\ -0.02 \\ -0.57 \\ 0.06 \\ -0.04 \\ -0.15 \end{bmatrix}, \quad (40)$$

$$u_5 = \begin{bmatrix} -0.12 \\ -0.72 \\ -0.07 \\ 0.65 \\ -0.18 \\ 0.09 \end{bmatrix}, \quad u_6 = \begin{bmatrix} 0.02 \\ 0.29 \\ 0.28 \\ 0.47 \\ 0.03 \\ -0.78 \end{bmatrix},$$

respectively. The variance contribution rates are

$$\begin{aligned} \eta_1 &= 46\%; & \eta_2 &= 20\%; & \eta_3 &= 16\%; \\ \eta_4 &= 14\%; & \eta_5 &= 4\%; & \eta_6 &= 0\%. \end{aligned} \quad (41)$$

Summing up η_q 's, we obtain

$$\begin{aligned} \eta_\Sigma(1) &= 46\%; & \eta_\Sigma(2) &= 65\%; & \eta_\Sigma(3) &= 81\%; \\ \eta_\Sigma(4) &= 95\%; & \eta_\Sigma(5) &= 100\%; & \eta_\Sigma(6) &= 100\%. \end{aligned} \quad (42)$$

A Pareto analysis chart is used to compare the percentage of variability explained by each principal component (see Figure 3). There is a clear break in the amount of variance accounted for by each component between the first and second components. However, that component by itself explains less than 50% of the variance, so more components

TABLE 6: The illustrative example.

j	x_{j1}	x_{j2}	x_{j3}	x_{j4}	x_{j5}	x_{j6}	CT_j
1	24	1261	181	781	112	0.92	935
2	24	1263	181	762	127	0.90	958
3	24	1220	176	761	127	0.89	1047
4	23	1282	178	802	127	0.94	1011
5	23	1303	180	780	175	0.93	1068
6	23	1281	183	782	175	0.93	1143
7	23	1242	184	741	163	0.89	1103
8	24	1262	182	681	139	0.86	1250
9	22	1260	182	701	98	0.86	1181
10	22	1260	179	700	257	0.87	1194
11	24	1301	163	722	99	0.84	1260
12	22	1221	184	641	131	0.82	1240
13	23	1323	159	740	247	0.87	1180
14	24	1362	181	782	191	0.95	1227
15	24	1261	181	762	219	0.91	1236
16	23	1321	177	801	219	0.96	1215
17	22	1343	180	822	219	0.97	1228
18	24	1321	177	762	54	0.93	1266
19	25	1343	179	781	54	0.96	1285
20	25	1300	180	740	54	0.92	1272
21	22	1320	181	721	54	0.91	1310
22	24	1321	182	742	49	0.92	1265
23	23	1262	165	680	201	0.80	1308
24	22	1240	161	722	103	0.82	1331
25	23	1183	183	661	53	0.82	1294
26	23	1282	184	701	53	0.88	1314
27	22	1202	177	680	248	0.84	1321
28	23	1202	178	681	248	0.85	1353
29	24	1202	185	701	82	0.86	1226
30	23	1202	158	721	98	0.81	1301
31	24	1343	181	760	67	0.94	1280
32	24	1381	185	801	67	0.97	1286
33	22	1362	156	780	67	0.91	1252
34	23	1282	179	782	223	0.92	1214
35	23	1320	180	782	176	0.93	1251
36	25	1340	176	801	462	0.97	1222
37	23	1320	182	781	168	0.95	1187
38	22	1361	181	781	141	0.94	1205
39	22	1381	179	781	95	0.97	1120
40	23	1363	178	802	179	0.97	1133

TABLE 7: New inputs to the FCM-FBPN.

z_{j1}	z_{j2}	z_{j3}
-0.56	0.91	-0.19
-0.13	0.87	-0.34
0.51	0.57	-0.37
-0.97	-0.10	0.20
-0.87	-0.20	-0.26
-0.75	0.14	-0.51
0.57	0.56	-0.66
1.30	1.18	-0.55
1.55	0.31	0.47
1.37	-0.87	-1.04
1.11	-0.59	0.91
3.04	0.63	-0.20
0.51	-2.44	-0.02
-1.94	0.12	-0.43
-0.30	0.35	-1.29
-1.62	-0.84	-0.48
-2.04	-1.24	-0.17
-0.87	0.77	0.89
-1.92	1.34	0.64
-0.58	1.70	0.34
0.22	0.23	1.29
-0.62	1.31	0.73
2.54	-1.26	-0.16
2.39	-1.64	1.20
3.02	1.57	0.14
0.89	1.21	0.66
2.56	-0.74	-1.19
2.19	-0.13	-1.54
1.61	1.90	-0.42
2.72	-1.23	0.87
-1.27	0.99	0.71
-2.56	1.07	0.78
-0.37	-2.44	2.47
-0.60	-0.51	-0.82
-1.06	-0.27	-0.17
-2.54	-1.36	-3.41
-1.31	-0.02	-0.18
-1.32	-0.63	0.67
-1.77	-0.58	1.32
-2.13	-0.66	0.12

are considered. To meet the requirement $\eta_{\Sigma}(p) \geq 85\% \sim 90\%$, p is chosen as 3. We can see that the first three principal components explain roughly 80% of the total variability in the standardized data, so we reduce the issue to these three components in order to visualize the data.

The component scores are recalculated as the new inputs to the FCM-FBPN (see Table 7). These scores contain the coordinates of the original data in the new coordinate system defined by the principal components.

Subsequently, jobs are classified using FCM based on the new variables. The results of the S test are summarized in Table 8. In this case, the optimal number of job categories is 5.

TABLE 8: The results of the S test.

Number of categories (K)	J_m	$e_{\min}^2$	S
2	1.96	0.14	0.34
3	1.21	0.09	0.34
4	0.86	0.07	0.30
5	0.67	0.06	0.26
6	0.53	0.03	0.43

However, there will be some categories with very few jobs. For this reason, the second best solution is used, that is, 4 categories. A common practice is to set a threshold of member-

FIGURE 3: The Pareto analysis chart.

FIGURE 4: The forecasting results from the FBPN.

ship μ_L to determine whether a job belongs to each category. With the decrease in the threshold, each category contains more jobs. Because the overall forecasting performance is affected by the outliers, μ_L should not be set too high. In this case, μ_L is set to 0.3. The classifying results are shown in Table 9.

After job classification, 3/4 of the examples in each category are used as the training example. The remaining 1/4 is left for testing. A three-layer FBPN is then used to predict the cycle time of jobs in each category according to the new variables with the following settings.

Single hidden layer

The number of neurons in the hidden layer: $2 * 3 = 6$

Convergence criterion: $SSE < 10^{-6}$ or 10000 epochs have been run

Iterations: 5

TABLE 9: The classifying results ($\mu_L = 0.3$).

Category	Jobs
1	4-6, 14-17
2	10-11, 13, 23-24, 27, 30, 33
3	1, 18-22, 31-32
4	2-3, 7-9, 12, 25-26, 28-29, 34-40

The forecasting results are shown in Figure 4. The performance level of the proposed methodology is compared with those of statistical analysis (i.e., multiple linear regression), CART, PCA-CART, BPN, FCM-BPN, and PCA-BPN in Table 10. The forecasting precision is evaluated with the average range:

$$\text{average range} = \frac{\sum_{j=1}^n |o_{j3} - o_{j1}|}{n}. \quad (43)$$

The average range of a nonbiased crisp approach can be measured with 6σ (or 6 RMSE):

$$\sigma^2 = \text{RMSE}^2 \frac{n}{n-k-1}, \quad (44)$$

where

$$\text{RMSE} = \sqrt{\frac{\sum_{j=1}^n (a_j - f_j)^2}{n}}. \quad (45)$$

k is the number of independent variables; a_j and f_j denote the actual value and forecast of job j , respectively; n is the total number of data. When $n \rightarrow \infty$, $\sigma^2 \approx \text{RMSE}^2$. Theoretically, the probability that such an interval contains the actual value is about 99.7%, under the assumption that residuals follow a normal distribution. The forecasting accuracy is measured with MAE, MAPE, and RMSE:

$$\text{MAE} = \frac{\sum_{j=1}^n |a_j - f_j|}{n}, \quad (46)$$

$$\text{MAPE} = \frac{\sum_{j=1}^n |a_j - f_j| / a_j}{n}.$$

The CART algorithm is based on that of Breiman et al. [42]. The results of CART are shown in Figure 5. In PCA-CART, the new variables found by PCA are used as input variables to determine the job cycle time. The results of PCA-CART are shown in Figure 6. The adjustment of the lower and upper bounds for the proposed methodology is shown in Figure 7.

According to the experimental results, we have the following.

- (1) The nonlinear nature of this problem is obvious, since the performance of statistical analysis is poor, and statistical analysis is a linear approach.
- (2) The accuracy of the job cycle time forecasting by the proposed fuzzy data mining approach is better than those of the compared approaches because the proposed approach has achieved 25%~85% reductions in MAPE.

TABLE 10: Comparison of the forecasting performance levels.

Category	Average range	MAE (hrs)	MAPE	RMSE (hrs)
Statistical analysis	594	73	6.1%	99
CART	438	50	4.2%	73
PCA-CART	294	36	3.1%	49
BPN	414	30	2.4%	69
FCM-BPN	228	15	1.2%	38
PCA-BPN	402	27	2.3%	67
The proposed methodology	169	11	0.9%	29

FIGURE 5: Results of CART.

- (3) Moreover, the precision level of the proposed methodology, measured in terms of the average range of forecasts, is also superior to those of the compared approaches. The proposed methodology can be used to specify a very narrow range of the job cycle time. Since the internal due date is based on the cycle time forecast, a narrow range promotes the reliability of the internal due date.
- (4) The proposed method is unusually slow in its modification of the lower bound (see Figure 7); a more effective way to solve this problem is needed.
- (5) PCA seems to be useful for methods with job classification like CART. The combination of PCA and CART has significantly better forecasting accuracy than either method in isolation.
- (6) The simple combination of PCA and BPN does not have much effect. The main effect of PCA is to improve the correctness of the job classification.

In order to test the performance of the proposed methodology on real world situations, it has been applied to schedule a problem of 457 jobs from a wafer fabrication factory located in Taichung City Scientific Park, Taiwan. There are more than ten products in the wafer fabrication factory. The wafer fabrication factory has a monthly capacity of 20,000 wafers. Jobs are regularly released into the wafer fabrication factory. Various types of priorities are assigned to these jobs. Each

FIGURE 6: Results of PCA-CART.

product has 150~200 steps and six to nine reentries to the most bottlenecked machine. The first-in first-out (FIFO) rule is used to dispatch the jobs. The production characteristic of “reentry,” which is highly relevant to the semiconductor industry, is clearly reflected in this problem. It also shows the difficulties facing production planners and schedulers who attempt to provide an accurate due date for a product with a very complicated routing. For each job, twelve parameters were collected or retrieved from production management information system (PROMIS) databases and reports. After a backward elimination by regression analysis, six parameters (the job size, factory utilization, the queue length on the route, the queue length before the bottleneck, the work-in-process (WIP), and the average waiting time) were chosen as the inputs to the FBP. Here “the average waiting time” refers to the average of the waiting times of the three jobs that were completed most recently, not to the waiting time of the job under consideration. The proposed methodology and several existing methods have been applied to this case. The forecasting precision (measured in terms of the average range) and accuracy (measured in terms of RMSE) of these approaches are compared in Figures 8 and 9, respectively. The samples in this case exhibited notable variability, which limited the performance of the forecasting methods. Nevertheless, it is obvious that the proposed methodology still outperformed the six existing methods in both regards. It is also noteworthy that the methods based on job classification (such as CART, PCA-CART, FCM-BPN, and the proposed methodology) achieved better forecasting accuracy than methods without job classification. Such an advantage was further strengthened by variable replacement techniques like PCA.

4. Conclusions and Directions for Future Research

Job cycle time forecasting is an important task in any wafer fabrication factory. In this field, many data mining methods

FIGURE 7: The adjustment of the lower bounds.

FIGURE 8: Comparison of the forecasting precision.

FIGURE 9: Comparison of the forecasting precision.

have been applied. A fuzzy data mining approach for precise and accurate job cycle time forecasting is presented in this study. The forecast of the job cycle time by the proposed methodology is a fuzzy value; to improve precision, an effective procedure tightens the upper and lower bounds of the forecast. The forecasting accuracy is enhanced by a combination of PCA, FCM, and FBPN.

This study presents an illustrative example and a real case with the data of 457 jobs from a wafer fabrication factory. According to the experimental results, we have the following.

- (1) the forecasting accuracy (measured with MAE, MAPE, and RMSE) of the PCA-FCM-FBPN was significantly better than the accuracy levels of the other approaches.
- (2) Methods based on job classification have better forecasting accuracy. The incorporation of PCA makes the relationships between variables clearer and strengthens the advantage of job classification.
- (3) It is possible to specify a very narrow range of the job cycle time using the proposed methodology.

The upper bound can be adjusted faster than the lower bound. An effective method is needed to tackle this issue. In addition, the FBPN approach focuses on the modification of the threshold on the output node. Similar concepts can be applied to modify other parameters in the FBPN.

Acknowledgment

This study was financially supported by the National Science Council of Taiwan.

References

- [1] P. Backus, M. Janakiram, S. Mowzoon, G. C. Runger, and A. Bhargava, "Factory cycle-time prediction with a data-mining approach," *IEEE Transactions on Semiconductor Manufacturing*, vol. 19, no. 2, pp. 252–258, 2006.
- [2] W. L. Pearn, S. H. Chung, and C. M. Lai, "Due-date assignment for wafer fabrication under demand variate environment," *IEEE Transactions on Semiconductor Manufacturing*, vol. 20, no. 2, pp. 165–175, 2007.
- [3] T. Chen, A. Jeang, and Y.-C. Wang, "A hybrid neural network and selective allowance approach for internal due date assignment in a wafer fabrication plant," *International Journal of Advanced Manufacturing Technology*, vol. 36, no. 5-6, pp. 570–581, 2008.
- [4] C. F. Chien, C. Y. Hsu, and C. W. Hsiao, "Manufacturing intelligence to forecast and reduce semiconductor cycle time," *Journal of Intelligent Manufacturing*, pp. 1–14, 2011.
- [5] T. Chen, "A fuzzy back propagation network for output time prediction in a wafer fab," *Applied Soft Computing Journal*, vol. 2, no. 3, pp. 211–222, 2003.
- [6] Y. Meidan, B. Lerner, G. Rabinowitz, and M. Hassoun, "Cycle-time key factor identification and prediction in semiconductor manufacturing using machine learning and data mining," *IEEE Transactions on Semiconductor Manufacturing*, vol. 24, no. 2, pp. 237–248, 2011.
- [7] L.-X. Wang and J. M. Mendel, "Generating fuzzy rules by learning from examples," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 22, no. 6, pp. 1414–1427, 1992.
- [8] P.-C. Chang, J.-C. Hieh, and T. W. Liao, "Evolving fuzzy rules for due-date assignment problem in semiconductor manufacturing factory," *Journal of Intelligent Manufacturing*, vol. 16, no. 4-5, pp. 549–557, 2005.
- [9] P.-C. Chang and J. C. Hsieh, "A neural networks approach for due-date assignment in a wafer fabrication factory," *International Journal of Industrial Engineering*, vol. 10, no. 1, pp. 55–61, 2003.
- [10] D. Y. Sha and S. Y. Hsu, "Due-date assignment in wafer fabrication using artificial neural networks," *International Journal of Advanced Manufacturing Technology*, vol. 23, no. 9-10, pp. 768–775, 2004.
- [11] T. Chen, "An intelligent hybrid system for wafer lot output time prediction," *Advanced Engineering Informatics*, vol. 21, no. 1, pp. 55–65, 2007.
- [12] T. Beeg, "Wafer fab cycle time forecast under changing loading situations," in *Proceedings of the IEEE Conference and Workshop Advanced Semiconductor Manufacturing (ASMC '04)*, pp. 339–343, May 2004.
- [13] T. Chen, Y. C. Wang, and H. R. Tsai, "Lot cycle time prediction in a ramping-up semiconductor manufacturing factory with a SOM-FBPN-ensemble approach with multiple buckets and partial normalization," *International Journal of Advanced Manufacturing Technology*, vol. 42, no. 11-12, pp. 1206–1216, 2009.
- [14] T. Chen, H. C. Wu, and Y. C. Wang, "Fuzzy-neural approaches with example post-classification for estimating job cycle time in a wafer fab," *Applied Soft Computing Journal*, vol. 9, no. 4, pp. 1225–1231, 2009.
- [15] I. Tirkel, "Cycle time prediction in wafer fabrication line by applying data mining methods," in *Proceedings of the 22nd Annual IEEE/SEMI Advanced Semiconductor Manufacturing Conference (ASMC '11)*, p. 5, Saratoga Springs, NY, USA, May 2011.
- [16] Y. T. Tai, W. L. Pearn, and J. H. Lee, "Cycle time estimation for semiconductor final testing processes with Weibull-distributed waiting time," *International Journal of Production Research*, vol. 50, no. 2, pp. 581–592, 2012.
- [17] A. I. Sivakumar and C. S. Chong, "A simulation based analysis of cycle time distribution, and throughput in semiconductor backend manufacturing," *Computers in Industry*, vol. 45, no. 1, pp. 59–78, 2001.

- [18] H. Okumura, "Simulation analysis of relation between tool variability and cycle time," in *Proceedings of the International Symposium on Semiconductor Manufacturing (ISSM) and e-Manufacturing and Design Collaboration Symposium (eMDC '11)*, pp. 1–10, Hsinchu, Taiwan, 2011.
- [19] C. P. L. Veeger, L. F. P. Etman, E. Lefeber, I. J. B. F. Adan, J. Van Herk, and J. E. Rooda, "Predicting cycle time distributions for integrated processing workstations: an aggregate modeling approach," *IEEE Transactions on Semiconductor Manufacturing*, vol. 24, no. 2, pp. 223–236, 2011.
- [20] P. C. Chang, J. C. Hsieh, and T. W. Liao, "A case-based reasoning approach for due date assignment in a wafer fabrication factory," in *Proceedings of the 4th International Conference on Case-Based Reasoning (ICCBR '01)*, vol. 2080 of *Lecture Notes in Computer Science*, pp. 648–659, 2001.
- [21] C. Chiu, P. C. Chang, and N. H. Chiu, "A case-based expert support system for due-date assignment in a wafer fabrication factory," *Journal of Intelligent Manufacturing*, vol. 14, no. 3–4, pp. 287–296, 2003.
- [22] P.-C. Chang, Y.-W. Wang, and C.-H. Liu, "Combining SOM and GA-CBR for flow time prediction in semiconductor manufacturing factory," in *Rough Sets and Current Trends in Computing*, vol. 4259 of *Lecture Notes in Computer Science*, pp. 767–775, Springer, Berlin, Germany, 2006.
- [23] T. Chen, "A hybrid look-ahead SOM-FBPN and FIR system for wafer-lot-output time prediction and achievability evaluation," *International Journal of Advanced Manufacturing Technology*, vol. 35, no. 5–6, pp. 575–586, 2007.
- [24] T. Chen, "A SOM-FBPN-ensemble approach with error feedback to adjust classification for wafer-lot completion time prediction," *International Journal of Advanced Manufacturing Technology*, vol. 37, no. 7–8, pp. 782–792, 2008.
- [25] T. Chen, "Job cycle time estimation in a wafer fabrication factory with a bi-directional classifying fuzzy-neural approach," *International Journal of Advanced Manufacturing Technology*, vol. 56, no. 9, pp. 1007–1018, 2011.
- [26] P. C. Chang and T. W. Liao, "Combining SOM and fuzzy rule base for flow time prediction in semiconductor manufacturing factory," *Applied Soft Computing Journal*, vol. 6, no. 2, pp. 198–206, 2006.
- [27] T. Chen, "A hybrid SOM-BPN approach to lot output time prediction in a wafer fab," *Neural Processing Letters*, vol. 24, no. 3, pp. 271–288, 2006.
- [28] T. Chen, "Predicting wafer-lot output time with a hybrid FCM-FBPN approach," *IEEE Transactions on Systems, Man, and Cybernetics B*, vol. 37, no. 4, pp. 784–793, 2007.
- [29] T. Chen, "An intelligent mechanism for lot output time prediction and achievability evaluation in a wafer fab," *Computers and Industrial Engineering*, vol. 54, no. 1, pp. 77–94, 2008.
- [30] T. Chen, "Semiconductor unit cost forecasting with fuzzy collaborative forecasting models," *Algorithms*, vol. 5, no. 4, pp. 449–468, 2012.
- [31] Q. Song and B. S. Chissom, "Forecasting enrollments with fuzzy time series—part II," *Fuzzy Sets and Systems*, vol. 62, no. 1, pp. 1–8, 1994.
- [32] W. Pedrycz, "Collaborative architectures of fuzzy modeling," in *Computational Intelligence: Research Frontiers*, vol. 5050 of *Lecture Notes in Computer Science*, pp. 117–139.
- [33] T. Chen and Y. C. Wang, "Incorporating the FCM-BPN approach with nonlinear programming for internal due date assignment in a wafer fabrication plant," *Robotics and Computer-Integrated Manufacturing*, vol. 26, no. 1, pp. 83–91, 2010.
- [34] X. L. Xie and G. Beni, "A validity measure for fuzzy clustering," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 13, no. 8, pp. 841–847, 1991.
- [35] H. Nomura, I. Hayashi, and N. Wakami, "A learning method of fuzzy inference rules by descent method," in *Proceedings of the IEEE International Conference on Fuzzy Systems*, pp. 203–210, March 1992.
- [36] C.-J. Lin, "A GA-based neural fuzzy system for temperature control," *Fuzzy Sets and Systems*, vol. 143, no. 2, pp. 311–333, 2004.
- [37] J. Watada, H. Tanaka, and T. Shimomura, "Identification of learning curve based on possibilistic concepts," in *Applications of Fuzzy Set Theory in Human Factors*, Elsevier, The Netherlands, 1986.
- [38] D. Huang, T. Chen, and M.-J.J. Wang, "A fuzzy set approach for event tree analysis," *Fuzzy Sets and Systems*, vol. 118, no. 1, pp. 153–165, 2001.
- [39] H. Akaike, "A new look at the statistical model identification," *IEEE Transactions on Automatic Control*, vol. AC-19, pp. 716–723, 1974.
- [40] J. Rissanen, "Stochastic complexity and modeling," *The Annals of Statistics*, vol. 14, no. 3, pp. 1080–1100, 1986.
- [41] T. Chen and Y. C. Lin, "A collaborative fuzzy-neural approach for internal due date assignment in a wafer fabrication plant," *International Journal of Innovative Computing, Information and Control*, vol. 7, no. 9, pp. 5193–5210, 2011.
- [42] L. Breiman, J. Friedman, C. J. Stone, and R. A. Olshen, *Classification and Regression Trees*, Chapman & Hall, Boca Raton, Fla, USA, 1993.