

Research Article

An Iterative Procedure for Optimizing the Performance of the Fuzzy-Neural Job Cycle Time Estimation Approach in a Wafer Fabrication Factory

Toly Chen and Yi-Chi Wang

Department of Industrial Engineering and Systems Management, Feng Chia University, 100 Wenhwa Road, Seatwen, Taichung 408, Taiwan

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

Received 28 October 2012; Accepted 27 December 2012

Academic Editor: Peng Shi

Copyright © 2013 T. Chen and Y.-C. Wang. 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.

Estimating the cycle time of each job in a wafer fabrication factory is a critical task to every wafer manufacturer. In recent years, a number of hybrid approaches based on job classification (either preclassification or postclassification) for cycle time estimation have been proposed. However, the problem with these methods is that the input variables are not independent. In order to solve this problem, principal component analysis (PCA) is considered useful. In this study, a classifying fuzzy-neural approach, based on the combination of PCA, fuzzy c-means (FCM), and back propagation network (BPN), is proposed to estimate the cycle time of a job in a wafer fabrication factory. Since job classification is an important part of the proposed methodology, a new index is proposed to assess the validity of the classification of jobs. The empirical relationship between the S value and the estimation performance is also found. Finally, an iterative process is employed to deal with the outliers and to optimize the overall estimation performance. A real case is used to evaluate the effectiveness of the proposed methodology. Based on the experimental results, the estimation accuracy of the proposed methodology was significantly better than those of the existing approaches.

1. Introduction

The competition in the semiconductor industry has been very intense. How to obtain and maintain the competitive edge is an important task for all manufacturers in this industry. Quick response and on-time delivery are obviously pressing needs for any modern enterprise. To this end, accurate estimating and shortening the cycle time (flow time or manufacturing lead time) of each job in the factory is a prerequisite [1–9]. In a wafer fabrication factory, a job is usually composed of about 25 pieces of wafers and has hundreds of steps to be processed. In addition, a job may visit the same workstation more than once because the same operation may be needed multiple times. A wafer fabrication factory is therefore classified as a complicated reentrant production system.

Estimating the cycle time of each job in a wafer fabrication factory is very important to the factory because it can

signal the manager if the orders are progressed as they were expected. For example, if the estimated cycle time of a job is longer than as it was expected, then this order may not be completed to the customer before its due date. Some production control actions should then immediately be taken to accelerate the progress of the job [10]. That is why this paper studies the estimation of job cycle time in a wafer fabrication factory.

The existing approaches for the job cycle time estimation in a wafer fabrication factory can be classified into six categories: statistical analysis, production simulation (PS), back propagation network (BPN), case-based reasoning (CBR), fuzzy modeling methods, and hybrid approaches [9]. Among the six approaches, statistical analysis is the easiest, quickest, and most prevalent in practical applications. Most of the statistical analyses used linear regression equations to estimate the job cycle time (e.g., [11, 12]). Pearn et al. [13] fitted the distribution of the waiting time of a job with

a gamma distribution and then used a linear equation to estimate the job cycle time. Recently, Chien et al. [14] used nonlinear regression equations instead and then found out the relationship between the estimation error and some factory conditions and job attributes with a BPN to further improve the estimation accuracy. The major disadvantage of statistical analysis is the lack of estimation accuracy [9]. Conversely, a huge amount of data and lengthy simulation time required are two disadvantages of PS. Nevertheless, theoretically PS is the most accurate job cycle time estimation approach if the simulation model is completely valid and is continuously updated.

Considering effectiveness (estimation accuracy) and efficiency (execution time) simultaneously, Chang et al. [8], Chang and Hsieh [15], and Sha and Hsu [16] estimated the cycle time of a job in a wafer fabrication factory using a BPN with a single hidden layer. A BPN is an effective tool in modeling complex physical systems described by sets of different equations for prediction, control, and design purposes. Compared with some statistical analysis approaches, the average estimation accuracy measured with root mean squared error (RMSE) was considerably improved with the BPNs. For example, an improvement of about 40% in RMSE was achieved in the study of Chang et al. [8]. Chen [17] incorporated the job releasing plan of the 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. On the other hand, much less time and fewer data are required with a BPN than with PS. Chen et al. [18] and Beeg [19] estimated the cycle time of a job in a ramping up wafer fabrication factory. In their studies, Chen et al. used a BPN-based method, while Beeg tried to find out the impact of utilization for the cycle time.

Chiu et al. [20] established an expert system based on CBR for the job cycle time estimation. To effectively consider the uncertainty in the job cycle time, fuzzy logic was used in a number of studies. For example, Chang et al. [8] modified the first step (i.e., partitioning the range of each input variable into several fuzzy intervals) of the fuzzy modeling method proposed by Wang and Mendel [21], called the WM method, with a simple genetic algorithm (GA) and proposed the evolving fuzzy rule (EFR) approach to estimate the cycle time of a job in a wafer fabrication factory. Their EFR approach outperformed CBR and BPN in the estimation accuracy. Chen [9] constructed a fuzzy back propagation network (FBPN) that incorporated expert opinions to modify the inputs of the FBPN. Chen’s FBPN surpassed the crisp BPN especially with respect to efficiency.

In recent years, a number of hybrid approaches have been proposed, most of which classified jobs before estimating the cycle times. For example, Chen [7] combined self-organization map (SOM) and WM, in which jobs were classified using a SOM before estimating the cycle times of the jobs with WM. Chen and Wang [22] constructed a look-ahead k-means- (kM-) FBPN for the same purpose and discussed in detail the effects of using different look-ahead functions. More recently, Chen [17] proposed the look-ahead SOM-FBPN approach for the job cycle time estimation in a semiconductor factory [23]. Besides, a set of fuzzy inference

rules were also developed to evaluate the achievability of a cycle time forecast. Subsequently, Chen [24] added a selective allowance to the cycle time estimated using the look-ahead SOM-FBPN approach to determine the intermediate due date. Further, Chen et al. [23] showed that the suitability of combining the SOM and FBPN for the data could be improved with the feedback of the estimation error by the FBPN to adjust the classification results of the SOM. Chen et al. [25] proposed a postclassification fuzzy-neural approach in which a job was not pre-classified but rather postclassified after estimating the cycle time. Experimental results showed that the postclassification approach was better than the preclassification approaches in certain cases. In order to combine the advantages of preclassifying and post-classifying approaches, Chen [26] proposed a bi-directional classifying approach, in which jobs are not only pre-classified but also postclassified. Except few studies in which the historical data of a real semiconductor factory were collected, most studies in this field used simulated data [27].

In short, the followings have not done before:

- (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. However, this issue has rarely been addressed in previous studies of this field.
- (2) Job classification has been shown to be conducive to the estimation performance. However, most past studies chose classifiers subjectively and did not evaluate the performance of the classifier. Needless to say optimizing the classifier for the subsequent estimation task.

Principal component analysis (PCA) is a multivariate statistical analysis method. This method constructs a series of linear combinations of the original variables to form a new variable, so that these new variables are unrelated to each other as much as possible, and the relationship among them can be reflected in a better way. In this study, a fuzzy-neural approach, based on the combination of PCA, FCM, and BPN, is proposed to estimate the cycle time of a job in a wafer fabrication factory. The motivation of this study is explained as follows.

- (1) While in the past some studies combined PCA and FCM, the references on the combination of PCA, FCM, and BPN are still very limited. Chen [28] applied PCA to modify the inputs to a BPN for the job cycle time estimation. The estimation accuracy of PCA-BPN was slightly better than that of BPN alone. It seems that BPN can solve the dependencies of the input variables for the job cycle time estimation problems. PCA seems to be more important for the classification of jobs. This provides us with a motivation to improve the existing job cycle time estimation methods based on job classification.
- (2) FCM, as a part of the preclassifying approach, cannot be evaluated alone. Its success depends on the

performance of the subsequent estimation task. This provides us with a motivation to assess the validity of the classification of jobs from this point of view.

- (3) The S test is a commonly used method to determine the best number of categories in FCM. However, whether this way directly favors the estimation performance has not been confirmed.

The contribution compared with some previous works in the literature includes the following.

- (1) With factors that are dependent on each other, jobs may be misclassified if FCM is used alone. This may be harmful to the estimation accuracy of BPN, because incorrect examples are used to train the BPN. The fuzzy-neural approach replaces the original factors with new independent factors and is expected to be able to generate the correct classification results. The correctness of the classification results must be judged from the estimation performance. In order to measure that, two new indexes are defined.
- (2) It is anticipated that the new factors found out by PCA have a more explicit relationship with the job cycle time. As a result, the training of BPN may be accelerated. This also means that a more accurate relationship between the factors and the cycle time can also be generated with the same time.
- (3) A new index is proposed to assess the validity of the classification of jobs.
- (4) The empirical relationship between the S value and the estimation performance is found.
- (5) Outliers, that is, jobs that cannot be classified definitely, have not been dealt with properly in the past. However, the overall estimation performance is often affected by the outliers. For this reason, an iterative process is established in this study, which can optimize the overall estimation performance.

The differences between the proposed methodology and the previous methods are summarized in Table 1.

The remainder of this paper is organized as follows. Section 2 introduces the proposed PCA-FCM-BPN approach. An example is employed to illustrate the proposed methodology. A case with the real data from a wafer fabrication factory is investigated in Section 3. The performance of the proposed methodology is compared with those of the existing approaches for this real case. Based on the results, some points are made in analysis. Finally, the concluded remarks with a view to the future are given in Section 4.

2. Methodology

Two characteristics of the proposed methodology are input replacement and job classification. These features not are mathematical skills, but also have implications for the operations of a wafer fabrication factory. First, in the useful information for the estimation of the job cycle time, many factors are in fact mutually dependent. For example, it is

well known that the utilization of a factory increases when the work-in-process (WIP) level in the factory rises. Both utilization and the WIP level are important factors considered in some job cycle time estimation approaches. Whether the dependence of the factors will lead to problems in the classification of jobs needs to be checked. Therefore, the replacement of these factors with new independent variables is worth a try.

On the other hand, a number of job cycle time estimation approaches in this field classify jobs. A well-known concept is that the cycle time of a job is proportional to the WIP level of the factory, according to Little's law; however, that only holds when the factory utilization is 100%. Therefore, it is reasonable to divide jobs into two categories: jobs that are released into the factory when the factory utilization is 100% and jobs released when the factory utilization is less than 100%.

The architecture of the proposed methodology is shown in Figure 1.

2.1. Variable Replacement Using PCA. First, PCA is used to replace the inputs to the FCM-BPN. The combination of PCA and FCM has proven to be a more effective classifier than FCM alone [29]. PCA consists of the following steps.

- (1) Raw data standardization: to eliminate the difference between the dimensions and the impact of large numerical difference in the original variables, The original variables are standardized as the following:

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

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 :

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

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:

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

TABLE 1: The differences between the proposed methodology and the previous methods.

Method	SOM-WM [7], SOM-FBPN [17], kM-FBPN [22–24]	BPN-BPN [25]	FCM-FBPN-RBF [26]	The proposed methodology
Job preclassification	Yes	No	Yes	Yes
Job post/reclassification	No	Yes	Yes	Yes
Parameter replacement	No	No	No	Yes
Dealing with outliers	No	No	No	Yes
Iteration	No	No	No	Yes

*RBF is radial basis function network.

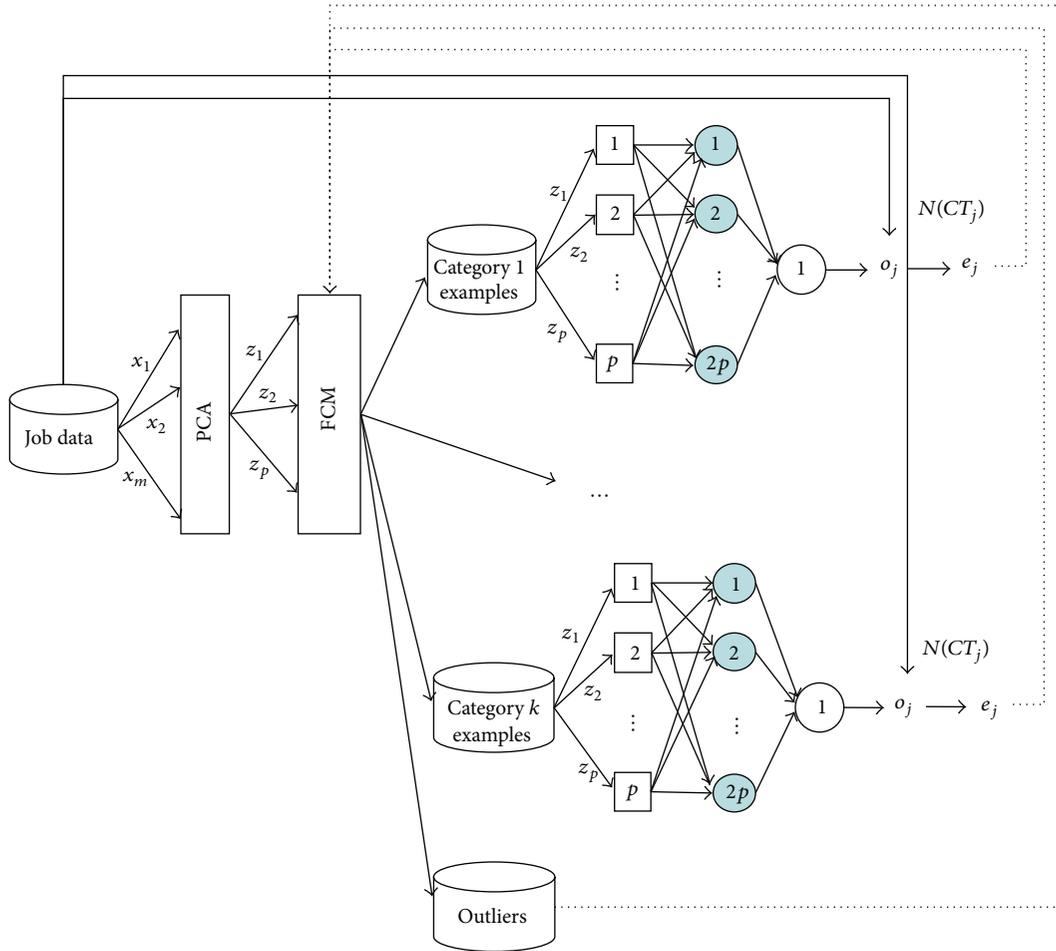


FIGURE 1: The architecture of the proposed methodology.

where $q = 1 \sim m$, and the accumulated variance contribution rate is

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

where $p = 1 \sim m$. Choose the smallest p value such that $\eta_{\Sigma}(p) \geq 85\% \sim 90\%$. A Pareto analysis chart can be used to compare the percent variability explained by each principal component.

(4) Formation of the following matrices:

$$\begin{aligned} U_{m \times p} &= [u_1, u_2, \dots, u_p], \\ Z_{n \times p} &= X_{n \times m}^* U_{m \times p}. \end{aligned} \quad (5)$$

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

TABLE 2: An example.

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

To illustrate the application of the proposed methodology, an example is given in Table 2. To get a quick impression of the data, a box plot is made in Figure 2. Note that there is substantially more variability in x_{j2} , x_{j4} , and x_{j5} than in the remaining variables.

Subsequently, we standardize the data (see Table 3) and obtain the correlation matrix as

$$R = \begin{bmatrix} 0.97 & 0.10 & 0.16 & 0.21 & -0.03 & 0.25 \\ 0.10 & 0.98 & 0.01 & 0.70 & -0.01 & 0.78 \\ 0.16 & 0.01 & 0.98 & 0.05 & -0.07 & 0.37 \\ 0.21 & 0.70 & 0.05 & 0.98 & 0.15 & 0.86 \\ -0.03 & -0.01 & -0.07 & 0.15 & 0.98 & 0.10 \\ 0.25 & 0.78 & 0.37 & 0.86 & 0.10 & 0.98 \end{bmatrix}. \quad (6)$$

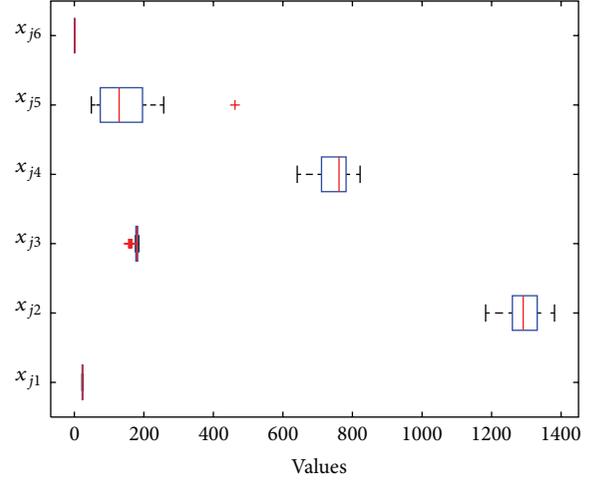


FIGURE 2: The box plot.

The eigenvalues and eigenvectors of R are calculated as the following:

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

$$\begin{aligned} u_1 &= \begin{bmatrix} 0.20 \\ 0.52 \\ 0.16 \\ 0.56 \\ 0.07 \\ 0.59 \end{bmatrix}, & u_2 &= \begin{bmatrix} 0.47 \\ -0.19 \\ 0.68 \\ -0.19 \\ -0.49 \\ 0.05 \end{bmatrix}, \\ u_3 &= \begin{bmatrix} -0.27 \\ 0.31 \\ -0.32 \\ 0.03 \\ -0.85 \\ -0.02 \end{bmatrix}, & u_4 &= \begin{bmatrix} 0.81 \\ -0.02 \\ -0.57 \\ 0.06 \\ -0.04 \\ -0.15 \end{bmatrix}, \\ u_5 &= \begin{bmatrix} -0.12 \\ -0.72 \\ -0.07 \\ 0.65 \\ -0.18 \\ 0.09 \end{bmatrix}, & u_6 &= \begin{bmatrix} 0.02 \\ 0.29 \\ 0.28 \\ 0.47 \\ 0.03 \\ -0.78 \end{bmatrix}, \end{aligned} \quad (7)$$

respectively. The variance contribution rates are

$$\begin{aligned} \eta_1 &= 46\%, & \eta_2 &= 20\%, & \eta_3 &= 16\%, \\ \eta_4 &= 14\%, & \eta_5 &= 4\%, & \eta_6 &= 0\%. \end{aligned} \quad (8)$$

TABLE 3: The standardized data.

j	x_{j1}	x_{j2}	x_{j3}	x_{j4}	x_{j5}	x_{j6}
1	0.88	-0.53	0.40	0.74	-0.40	0.37
2	0.88	-0.49	0.48	0.30	-0.22	-0.05
3	0.88	-1.27	-0.17	0.29	-0.22	-0.31
4	-0.22	-0.15	0.07	1.18	-0.22	0.65
5	-0.22	0.24	0.37	0.71	0.35	0.58
6	-0.22	-0.17	0.78	0.74	0.35	0.45
7	-0.22	-0.87	0.90	-0.14	0.21	-0.19
8	0.88	-0.51	0.53	-1.45	-0.08	-0.87
9	-1.32	-0.55	0.60	-1.01	-0.56	-0.81
10	-1.32	-0.54	0.23	-1.03	1.34	-0.55
11	0.88	0.20	-1.87	-0.57	-0.56	-1.19
12	-1.32	-1.25	0.80	-2.33	-0.18	-1.64
13	-0.22	0.60	-2.40	-0.16	1.22	-0.60
14	0.88	1.31	0.47	0.75	0.55	0.94
15	0.88	-0.53	0.40	0.31	0.88	0.19
16	-0.22	0.57	-0.05	1.17	0.88	1.13
17	-1.32	0.97	0.33	1.62	0.88	1.38
18	0.88	0.56	-0.01	0.31	-1.09	0.62
19	1.97	0.96	0.15	0.74	-1.09	1.06
20	1.97	0.19	0.38	-0.16	-1.09	0.36
21	-1.32	0.55	0.51	-0.57	-1.09	0.10
22	0.88	0.55	0.54	-0.13	-1.16	0.32
23	-0.22	-0.52	-1.59	-1.47	0.67	-2.00
24	-1.32	-0.91	-2.11	-0.57	-0.51	-1.54
25	-0.22	-1.95	0.72	-1.89	-1.11	-1.64
26	-0.22	-0.15	0.89	-1.01	-1.11	-0.36
27	-1.32	-1.59	-0.07	-1.47	1.23	-1.26
28	-0.22	-1.60	0.07	-1.47	1.23	-1.11
29	0.88	-1.60	0.93	-1.03	-0.76	-0.87
30	-0.22	-1.59	-2.57	-0.58	-0.56	-1.86
31	0.88	0.97	0.41	0.27	-0.94	0.70
32	0.88	1.65	1.00	1.16	-0.94	1.27
33	-1.32	1.30	-2.73	0.71	-0.94	0.10
34	-0.22	-0.14	0.16	0.74	0.92	0.33
35	-0.22	0.55	0.34	0.75	0.36	0.54
36	1.97	0.91	-0.23	1.17	3.79	1.28
37	-0.22	0.55	0.61	0.73	0.27	0.91
38	-1.32	1.28	0.42	0.72	-0.05	0.81
39	-1.32	1.66	0.14	0.72	-0.60	1.36
40	-0.22	1.33	0.13	1.18	0.40	1.42

Summing up η_q 's, we obtain the following

$$\begin{aligned}
 \eta_{\Sigma}(1) &= 46\%, & \eta_{\Sigma}(2) &= 65\%, \\
 \eta_{\Sigma}(3) &= 81\%, & \eta_{\Sigma}(4) &= 95\%, \\
 \eta_{\Sigma}(5) &= 100\%, & \eta_{\Sigma}(6) &= 100\%.
 \end{aligned} \tag{9}$$

A Pareto analysis chart is used to compare the percent 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 the second

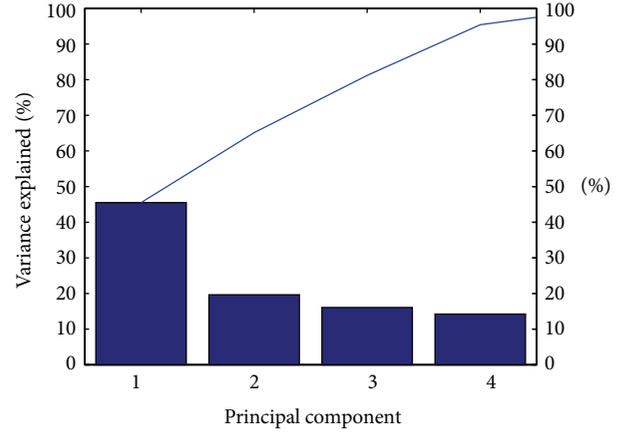


FIGURE 3: The Pareto analysis chart.

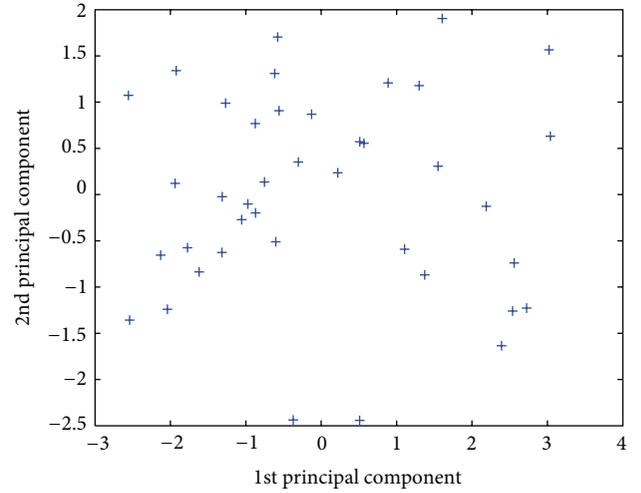


FIGURE 4: The component scores.

components. However, that component by itself can only explain less than 50% of the variance, so more components may be needed. 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 that might be a reasonable way to reduce the dimensions in order to visualize the data.

Subsequently, the component scores are computed (see Table 4), which contain the coordinates of the original data in the new coordinate system defined by the principal components and will be used as the new inputs to the FCM-BPN. In Figure 4, the first two columns of the component scores are plotted, showing the data projected onto the first two principal components.

2.2. Classifying Jobs Using FCM. After employing PCA, examples are then classified using FCM. If a crisp clustering method is applied instead, then it is very likely that some clusters will have very few examples. In contrast, an example belongs to multiple clusters to different degrees in FCM,

TABLE 4: New inputs to the FCM-BPN.

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

which provides a solution to this problem. Similarly, in probability theory, the naïve Bayes method provides the probability that the item belongs to each class. However, the application of FCM can consider subjective factors in classifying the jobs. Algorithm 1.

FCM classifies jobs by minimizing the following objective function:

$$\text{Min} \sum_{k=1}^K \sum_{j=1}^n \mu_{j(k)}^m e_{j(k)}^2, \quad (10)$$

where K is the required number of categories; n is the number of jobs; $\mu_{j(k)}$ indicates the membership that job j belongs to 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 described as follows.

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

$$\begin{aligned} \bar{z}_{(k)} &= \{\bar{z}_{(k)q}\}, \quad k = 1 \sim K, \\ \bar{z}_{(k)q} &= \frac{\sum_{j=1}^n \mu_{j(k)}^m z_{jq}}{\sum_{j=1}^n \mu_{j(k)}^m}, \quad k = 1 \sim K, \quad q = 1 \sim p, \\ \mu_{j(k)} &= \frac{1}{\sum_{g=1}^K (e_{j(k)}/e_{j(g)})^{2/(m-1)}}, \quad j = 1 \sim n, \quad k = 1 \sim K, \\ e_{j(k)} &= \sqrt{\sum_{q=1}^p (z_{jp} - \bar{z}_{(k)p})^2}, \quad j = 1 \sim n, \quad k = 1 \sim K, \end{aligned} \quad (11)$$

where $\bar{z}_{(k)}$ is the centroid of category k . $\mu_{j(k)}^{(t)}$ is the membership that job j belongs to 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):

$$\max_k \max_j |\mu_{j(k)}^{(t)} - \mu_{j(k)}^{(t-1)}| < d. \quad (12)$$

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

The performance of FCM is highly affected by the settings for the initial values, and therefore can be repeated multiple times in order to find the optimal solution. Finally, the separate distance test (S test) proposed by Xie and Beni [30] can be applied to determine the optimal number of categories K as follows:

$$\text{Min } S \quad (13)$$

subject to

$$\begin{aligned} J_m &= \sum_{k=1}^K \sum_{j=1}^n \mu_{j(k)}^m e_{j(k)}^2, \\ e_{\min}^2 &= \min_{k_1 \neq k_2} \left(\sum_{q=1}^p (\bar{z}_{(k_1)q} - \bar{z}_{(k_2)q})^2 \right), \quad (14) \\ S &= \frac{J_m}{n \times e_{\min}^2}, \\ K &\in \mathbb{Z}^+. \end{aligned}$$

```

A=[0.3857 7175 0.5381;...; 0.1614 0.4281 0.5803]
c=2
[center, U, obj_fun]=fcm(A, c);
Jm=min(obj_fun)
e2_min=9999;
for i=1:c
    for j=i+1:c
        e2_sum=0;
        for k=1:3
            e2_sum=e2_sum+(center(i, k)-center(j, k))^2;
        end
        if e2_sum<e2_min
            e2_min=e2_sum;
        end
    end
end
e2_min
S=min(Jm)/(40*e2_min)

```

ALGORITHM 1: The sample MATLAB code for the FCM approach.

TABLE 5: 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

TABLE 6: The classifying results ($\mu_L = 0.5$).

Category	Jobs
1	1, 2, 18, 19, 20, 22, 31, 32
2	3, 7, 8, 9, 12, 25, 26, 29
3	4, 5, 6, 14, 16, 17, 34, 35, 37, 38, 39, 40
4	10, 11, 23, 24, 27, 30

TABLE 7: The classifying results ($\mu_L = 0.3$).

Category	Jobs
1	1, 2, 18, 19, 20, 21, 22, 31, 32
2	2, 3, 7, 8, 9, 12, 25, 26, 28, 29
3	4, 5, 6, 14, 15, 16, 17, 33, 34, 35, 36, 37, 38, 39, 40
4	10, 11, 13, 23, 24, 27, 28, 30, 33

The K value minimizing S determines the optimal number of categories.

The Fuzzy Logic Toolbox of MATLAB can be used to implement the FCM approach. A sample code is shown in

In the illustrative example, the data have been standardized and therefore are not normalized again. The results of the S test are summarized in Table 5. In this case, the optimal number of job categories was 5. 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 membership μ_L to determine whether a job belongs to each category. For example, if $\mu_L = 0.5$, then the classifying results are shown in Table 6. With the decrease in the threshold, each category will contain more jobs. For example, if $\mu_L = 0.3$, then the classifying results are shown in Table 7. Such a property can solve the problem of an insufficient number of examples.

We also note that the classification results are very different according to the new variables, compared with the results based on the original variables. In other words, the results of FCM and PCA-FCM are not the same.

- (1) The optimal number of categories in FCM is 6, while that in PCA-FCM is 5.
- (2) If jobs are divided into four categories in these two methods, then the results are compared in Figure 5. Many jobs have been reclassified, which means that the misclassification problem has been resolved after variable replacement.

In Figure 5, there are also some outliers that cannot be classified into any category.

2.3. Estimating the Cycle Time Using BPN. Finally, the jobs/examples of a category are learned with the same BPN. Artificial neural networks have been proposed to solve a wide variety of problems usually characterized by sets of different equations. Although there have been some more advanced artificial neural networks, such as compositional pattern-producing network, cascading neural network, and dynamic neural network, a well-trained BPN with an optimized structure can still produce very good results. The configuration of the BPN is established as follows.

- (1) Inputs: the new factors determined by PCA associated with the j th example/job. These factors have to be partially normalized so that their values fall within [0.1, 0.9] [18].

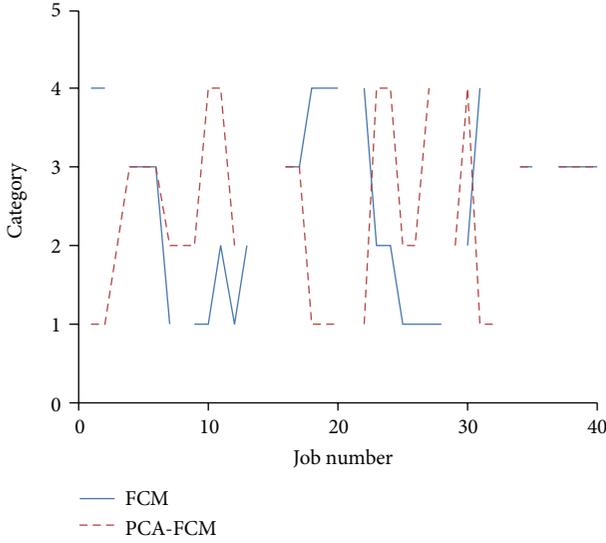


FIGURE 5: Comparison of the classification results by PCA-FCM and FCM.

- (2) Single hidden layer: generally one or two hidden layers are more beneficial for the convergence property of the BPN.
- (3) For simplicity, the number of neurons in the hidden layer is twice that 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. Such a phenomena is often called “over-fitting.” There has been some research considering the relation among the complexity of a BPN, the performance for the training data and the number of examples, for example using Akaike’s information criterion (AIC) or the minimum description length (MDL).
- (4) Output: the (normalized) cycle time estimate of the example.

The procedure for determining the parameter values is now described. After preclassification, a portion of the adopted examples in each category is fed as “training examples” into the BPN to determine the parameter values for the category. Two phases are involved at the training stage. At first, in the forward phase, inputs are multiplied with weights, summated, and transferred to the hidden layer. Then activated signals h_{jl} are outputted from the hidden layer as:

$$h_{jl} = \frac{1}{1 + e^{-n_{jl}^h}}, \quad (15)$$

where

$$\begin{aligned} n_{jl}^h &= I_{jl}^h - \theta_l^h, \\ I_{jl}^h &= \sum_{q=1}^p w_{ql}^h z_{jq} \end{aligned} \quad (16)$$

h_{jl} ’s are also transferred to the output layer with the same procedure. Finally, the output of the BPN is generated as:

$$o_j = \frac{1}{1 + e^{-n_j^o}}, \quad (17)$$

where

$$\begin{aligned} n_j^o &= I_j^o - \theta^o, \\ I_j^o &= \sum_{l=1}^L w_l^o h_{jl}. \end{aligned} \quad (18)$$

The output o_j is compared with the normalized step flow time $N(CT_j)$, for which RMSE is calculated as the following:

$$\text{RMSE} = \sqrt{\frac{\sum_{j=1}^n (o_j - N(CT_j))^2}{n}}. \quad (19)$$

Subsequently in the backward phase, some algorithms are applicable for training a BPN, such as 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 the 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 BPN is to minimize RMSE or equivalently the sum of squared error (SSE):

$$\text{SSE}(\beta) = \sum_{j=1}^n (N(CT_j) - f(\mathbf{x}_j, \beta))^2. \quad (20)$$

The Levenberg-Marquardt algorithm is an iterative procedure. In the beginning, the user should specify the initial values of the network parameters β . Let $\beta^T = (1, 1, \dots, 1)$ be a common practice. 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 $f(\mathbf{x}_j, \beta + \delta)$ that is approximated by its linearization as

$$f(\mathbf{x}_j, \beta + \delta) \approx f(\mathbf{x}_j, \beta) + \mathbf{J}_j \delta, \quad (21)$$

where

$$\mathbf{J}_j = \partial \frac{f(\mathbf{x}_j, \beta)}{\partial \beta} \quad (22)$$

is the gradient vector of f with respect to β . Substituting (21) into (20) leads to.

$$\text{SSE}(\beta + \delta) \approx \sum_{j=1}^n (N(CT_j) - f(\mathbf{x}_j, \beta) - \mathbf{J}_j \delta)^2. \quad (23)$$

```

tn_input=[0.843 0.831 ...; 0.839 0.859 ...; 0.9 0.9 ...; 0.878 0.889 ...; 0.875 0.858 ...;
0.822 0.827 ...]
tn_target=[0.849 0.849 ...]
net=newff([0 1; 0 1; 0 1; 0 1; 0 1; 0 1], [12, 1], {"logsig", "logsig"}, "trainlm");
net=init(net);
net.trainParam.show=10;
net.trainParam.lr=0.1;
net.trainParam.epochs=1000;
net.trainParam.goal=1e-4;
[net, tr]=train(net, tn_input, tn_target);
tn_output=sim(net, tn_input)
te_input=[0.825 0.844...; 0.824 0.835 ...; 0.9 0.9 ...; 0.878 0.889 ...; 0.883 0.875 ...;
0.807 0.820 ...]
te_output=sim(net, te_input)

```

ALGORITHM 2: The sample MATLAB code for the BPN approach.

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

$$(\mathbf{J}^T \mathbf{J}) \delta = \mathbf{J}^T (N(CT_j) - f(\mathbf{x}_j, \beta)), \quad (24)$$

where \mathbf{J} is the Jacobian matrix containing the first derivative of network error with respect to the weights and biases. Equation (24) includes a set of linear equations that can be solved for δ .

In the illustrative example, 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 BPN is then used to estimate the cycle time of jobs in each category according to the new variables with the following setting.

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.

For an outlier, the BPNs of all categories are applied to estimate the cycle time. The Neural Network Toolbox of MATLAB is used to implement the BPN approach. The sample code is shown in Algorithm 2. The estimation accuracy can be evaluated with mean absolute error (MAE), mean absolute percentage error (MAPE), and RMSE. The estimation performances are summarized in Table 8.

Obviously, the overall estimation performance is affected by the outliers. If the outliers can be dealt with properly, the overall estimation will be improved. To this end, an iterative feedback control procedure is established in the next subsection (see Figure 6), which can optimize the overall estimation performance. In the literature, there have been a few control mechanisms for various types of fuzzy systems [31–39]. On the other hand, we also compare the performances of the gradient descent algorithm and the Levenberg-Marquardt algorithm, as shown in Table 9.

TABLE 8: The estimation performances.

Category	MAE (hrs)	MAPE	RMSE (hrs)
1	18	1.6%	42
2	1	0.1%	3
3	6	0.4%	12
4	23	1.8%	53
Outliers	149	11.9%	152
Total	42	3.5%	85

TABLE 9: Comparing the performances of two training algorithms (group 4, convergence criterion: SSE < 10^{-6}).

Algorithm	Number of epochs	MAE	MAPE	RMSE
Gradient descent	10000	79	6.7%	98
Levenberg-Marquardt	<100	23	1.8%	53

2.4. Iterative Optimization

2.4.1. The Effectiveness of the S Test. Job classification in the proposed methodology is based on the combination of FCM (or PCA-FCM) and the S test, according to which the best number of categories is chosen. This classification method takes into account only the similarity of the parameters of jobs. Whether it has a decisive impact for the subsequent cycle time estimation is not clear. For this reason, the cycle time estimation performances with different numbers of categories are compared to verify the results from the S test. The results are shown in Figure 7. Y-axis is provided in a logarithmic scale to make the relationship clearer. Clearly, when the S value becomes smaller, the estimation error (in terms of MAPE) is also reduced. Therefore, choosing the clustering results with the smallest S value is helpful to the estimation accuracy.

2.4.2. The Correctness of Job Classification. There are absolute rules for the classification of jobs in a wafer fabrication factory. It usually depends on the purpose of job classification, apparently to enhance the estimation accuracy in the

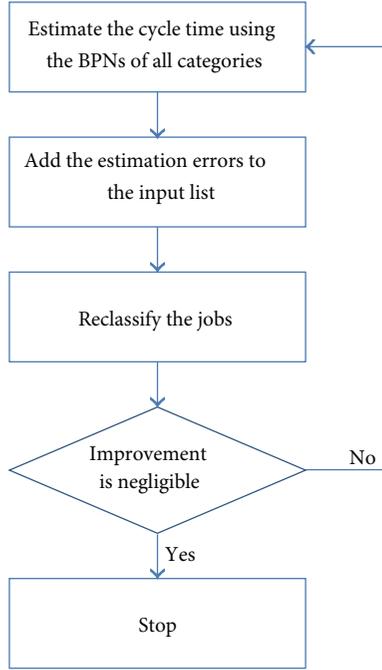


FIGURE 6: The iterative process of dealing with outliers.

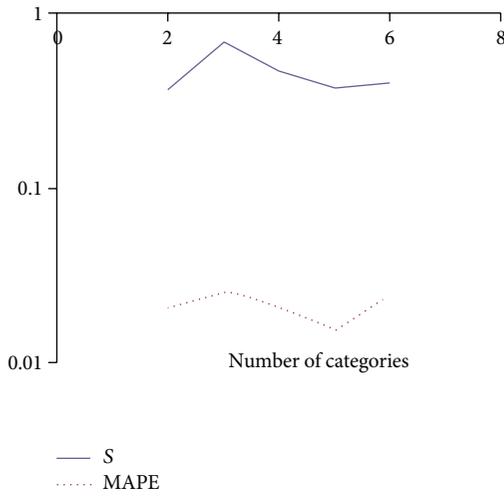


FIGURE 7: The relationship between the S value and MAPE.

proposed methodology. Therefore, a job is correctly classified if its cycle time is accurately estimated after classification. Otherwise, the job is misclassified.

Definition 1 (job misclassification). Assuming the cycle time of job j estimated by the BPN of category k is indicated with $o_j(k)$. The category of job j determined by classifier f is $f(j)$. Then, job j is correctly classified if

$$\begin{aligned} & |o_j(f(j)) - N(CT_j)| \\ & \leq |o_j(k) - N(CT_j)|, \quad \text{for any } k \neq f(j). \end{aligned} \quad (25)$$

A strong requirement of inequality (25) is

$$|o_j(f(j)) - N(CT_j)| \leq \min_{k \neq f(j)} |o_j(k) - N(CT_j)|, \quad (26)$$

while a weak requirement of this inequality is

$$|o_j(f(j)) - N(CT_j)| \leq \max_{k \neq f(j)} |o_j(k) - N(CT_j)|. \quad (27)$$

Definition 2 (the correctness of classifying a job). The degree that job j is correctly classified by classifier f is

$$\begin{aligned} \theta(f, j) &= \begin{cases} 1 & \text{if } |o_j(f(j)) - N(CT_j)| \leq \min_{k \neq f(j)} |o_j(k) - N(CT_j)| \\ 0 & \text{if } |o_j(f(j)) - N(CT_j)| \geq \max_{k \neq f(j)} |o_j(k) - N(CT_j)| \\ \frac{|o_j(f(j)) - N(CT_j)| - \max_{k \neq f(j)} |o_j(k) - N(CT_j)|}{\min_{k \neq f(j)} |o_j(k) - N(CT_j)| - \max_{k \neq f(j)} |o_j(k) - N(CT_j)|} & \text{otherwise.} \end{cases} \end{aligned} \quad (28)$$

Definition 3 (the correctness/correct percentage of the classification results). The correctness/correct percentage of the classification results by classifier f is

$$\theta(f) = \frac{\sum_{j=1}^n \theta(f, j)}{n} \cdot 100\%. \quad (29)$$

In the illustrative example, the correctness of job classification is evaluated, and the results are summarized in Table 10. In this example, the correctness of the classification results is 94%.

2.4.3. Feeding Back the Estimation Error and Reclassification. Subsequently, the estimation error is fed back to the FCM classifier to adjust the classification results. The difference with Chen and Wang's method [40] is that in the proposed methodology the BPNs of all categories are applied to estimate the cycle time of a job [41], and then the estimation errors arising from these BPNs all become additional inputs to the FCM, and jobs are reclassified. The new classification results are shown and compared with that before error feedback in Figure 8. After job reclassification, some outliers are assigned to the existing categories, and the overall estimation performance is improved in this way (see Table 11). The correctness of job classification $\theta(f)$ is now 97%. Job reclassification continues until the improvement in the overall estimation performance or in the correctness of job classification becomes negligible.

3. Further Comparisons

To further evaluate the advantages and/or disadvantages of the proposed methodology, eight existing approaches, statistical analysis, CBR [20], BPN, SOM-WM [7], EFR [21], SOM-FBPN [17], the postclassifying FBPN [25], and the bidirectional classifying BPN approach [26] were all applied

TABLE 10: The correctness of the classification results.

j	$\theta(f, j)$
1	1.00
2	1.00
3	1.00
4	1.00
5	1.00
6	1.00
7	1.00
8	1.00
9	1.00
10	1.00
11	1.00
12	1.00
13	1.00
14	1.00
15	1.00
16	1.00
17	1.00
18	1.00
19	1.00
20	1.00
21	1.00
22	1.00
23	1.00
24	1.00
25	1.00
26	0.63
27	1.00
28	0.58
29	1.00
30	1.00
31	1.00
32	1.00
33	1.00
34	1.00
35	1.00
36	1.00
37	1.00
38	0.55
39	0.00
40	1.00

to the collected data. Three performance measures including MAE, MAPE, and the minimal RMSE were evaluated.

The proposed methodology was implemented on a PC with an Intel Dual CPU E2200 2.2 GHz and 2.0 G RAM. FCM

TABLE 11: The estimation performances.

Category	MAE (hrs)	MAPE	RMSE (hrs)
1	1	0.0%	1
2	1	0.1%	2
3	1	0.1%	2
4	2	0.2%	5
Outliers	56	4.5%	80
Total	15	1.2%	36

TABLE 12: Comparisons of the performances of various approaches.

	MAE (hours)	MAPE	RMSE (hours)
Statistical analysis	73	6.1%	99
CBR	81	6.5%	104
BPN	33	2.8%	71
SOM-WM	30	2.5%	64
EFR	30	2.6%	65
SOM-FBPN	22	2.0%	38
Postclassifying FBPN	40	2.7%	88
Bidirectional classifying BPN	19	1.9%	37
The proposed methodology	15	1.2%	36

was implemented with the Fuzzy toolbox of MATLAB 2006a. In addition, BPN was implemented with the Neural Network Toolbox under the following conditions.

- (1) Number of epochs per replication: 10000.
- (2) Number of initial conditions/replications: 10.
- (3) Stop training if $MSE < 10^{-6}$ is satisfied or 10000 epochs have been run.

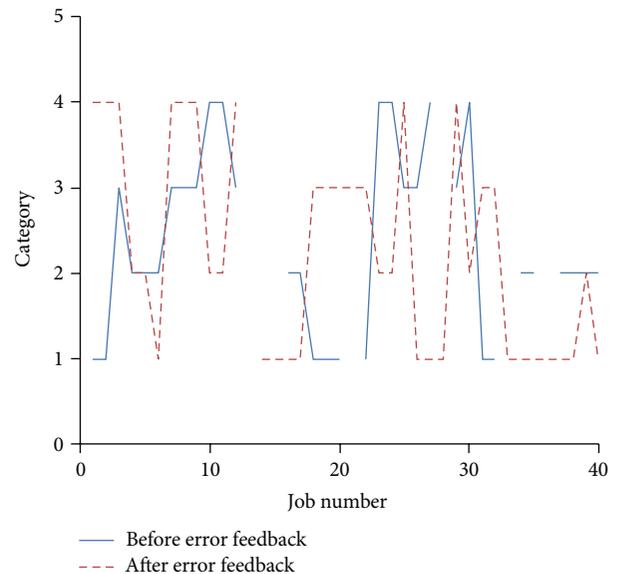


FIGURE 8: Comparison of the classification results.

Among the steps, PCA and FCM can be done instantaneously. The training of BPN usually takes less than 1 minute per replication.

The performances with the nine approaches are compared and summarized in Table 12.

In statistical analysis, a linear regression equation is used to estimate the job cycle time. In the CBR approach, the weights of factors (the cycle times of the previous cases) are proportional to the similarities of the new job with the previous cases. The optimal value of parameter k in the CBR approach was equal to the value that minimized the RMSE [8]. In the BPN approach, there was one hidden layer with 4~8 nodes, depending on the results of a preliminary analysis for establishing the best configuration. 3/4 of the collected data were used for training the BPN, while the remaining data were used for testing. In SOM-FBPN and SOM-WM, jobs were first classified with SOM. Subsequently, the examples of different categories were then learned with different FBPNs but with the same topology (or WM). In EFR, jobs are classified using fuzzy partition. In the post-classifying FBPN approach, a job was not pre-classified but rather post-classified after the estimation error has been generated. For this purpose, a BPN was used as the postclassification algorithm. In the bidirectional classifying approach, jobs are not only preclassified but also postclassified. The results of preclassification and postclassification are aggregated into a suitability index for each job. Each job is then assigned to the category to which its suitability index is the highest.

Statistical analysis was adopted as a comparison basis. According to experimental results, the following points are made.

- (1) The combination of BPN and PCA could reduce about 50% of space for storing the input variables in the modeling of the wafer fabrication system.
- (2) From the effectiveness viewpoint, the estimation accuracy (measured with the MAPE) of the proposed methodology was significantly better than those of the other approaches. The average advantage over statistical analysis is 80%.
- (3) The standard deviation of the cycle time for this case is 100 hours. Compared with this, the accuracy of the proposed methodology is good.
- (4) The estimation performance of the proposed methodology was also better than the existing classifying methods, such as SOM-WM, SOM-FBN, EFR, SOM-FBPN, the postclassifying FBPN, and the bidirectional classifying BPN approach. The advantage of the proposed methodology was reasonable due to the replacement of the variables and the iterative process of dealing with the outliers.
- (5) In general, the performances with the preclassifying approaches are better than that with the post-classifying approach.
- (6) The proposed methodology was also applied to other cases. The results are summarized in Table 13.

TABLE 13: Performances in other cases.

RMSE	Case I	Case II	Case III	Case IV
Statistical analysis	77	83	104	78
CBR	74	78	96	72
BPN	53	68	84	63
SOM-WM	49	71	86	66
EFR	50	50	62	47
SOM-FBPN	38	53	66	50
Postclassifying FBPN	62	93	113	86
Bidirectional classifying BPN	24	31	38	28
The proposed methodology	23	23	28	22

Wilcoxon signed-rank test [42] was then used to make sure whether or not the differences between the performance of the proposed methodology and those of the eight existing approaches are significant.

H_0 : When estimating the job cycle time the estimating performance of the proposed methodology is the same as that of the existing approach being compared.

H_1 : When estimating the job cycle time, the estimating performance of the proposed methodology is better than that of the existing approach being compared.

The results are summarized in Table 14. The null hypothesis H_0 was rejected at $\alpha = 0.05$, showing that the proposed methodology was superior to seven existing approaches in estimating the job cycle time.

- (7) To ascertain the effect of each treatment taken in the proposed methodology, the performances of BPN, FCM-BPN, PCA-BPN, and PCA-FCM-BPN (the proposed methodology) are compared in Table 15. Obviously, job classification (FCM) did contribute to the effectiveness of the proposed methodology, while the effect of variable replacement (PCA) was not obvious. The simultaneous application of the two treatments further improved the estimation accuracy for the testing data.

4. Conclusions and Directions for Future Research

Estimating the cycle time of each job in a wafer fabrication factory is a critical task to the wafer fabrication factory and has been widely studied in recent years. In order to further enhance the accuracy of the job cycle time estimation, PCA is applied to the FCM-BPN approach in this study, which is an innovative treatment in this field. Through replacing the variables, job classification can be more accurate. In addition, the relationship between the factors and the cycle time can be clearly specified.

On the other hand, since job classification is the core for the proposed methodology, a new index is used to validate the classification of jobs. The empirical relationship between the S value and the estimation performance is also found. Finally,

TABLE 14: Results of the Wilcoxon sign-rank test.

	H_0
Statistical analysis	$Z = 1.83^*$
CBR	1.83*
BPN	1.83*
SOM-WM	1.83*
EFR	1.64
SOM-FBPN	1.83*
Postclassifying FBPN	1.83*
Bidirectional classifying BPN	1.83*

* $P < 0.05$.** $P < 0.025$.*** $P < 0.01$.

TABLE 15: The effects of the treatments taken in the proposed methodology.

RMSE (hrs)	BPN	FCM-BPN	PCA-BPN	PCA-FCM-BPN
Training data	73	58	71	36

an iterative process is established to deal with the outliers to optimize the overall estimation performance.

An example is used to illustrate the proposed methodology. According to the experimental results,

- (1) the estimation accuracy (measured with MAE, MAPE, and RMSE) using the proposed methodology was significantly better than those with the existing approaches;
- (2) the advantage of PCA is for improving the correctness of job classification. The simple combination of PCA and BPN does not show much advantage;
- (3) after combining with PCA, the estimation accuracy of FCM-BPN was significantly improved;
- (4) the overall estimation performance is often affected by the outliers. The iterative procedure tries to remove the outliers and gradually improves the overall estimation performance.

Some other issues for this topic can be further investigated. Most of the existing methods are based on the job clustering. The aim of this study is to provide positive impacts on certain measures for these methods. However, if there are the other variable replacement techniques that can be as effective is also worth exploring in future studies. In addition, the iterative procedure used to optimize the results of job classification is quite time consuming especially for a large-scale problem, and therefore a more efficient way should be found.

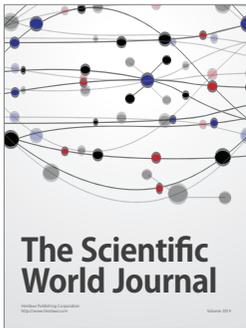
Acknowledgment

This work was supported by the National Science Council of Taiwan.

References

- [1] 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.
- [2] T. Chen, Y. C. Wang, and H. C. Wu, "A fuzzy-neural approach for remaining cycle time estimation in a semiconductor manufacturing factory: a simulation study," *International Journal of Innovative Computing, Information and Control*, vol. 5, no. 8, pp. 2125-2139, 2009.
- [3] T. Chen and Y. C. Lin, "A fuzzy back propagation network ensemble with example classification for lot output time prediction in a wafer fab," *Applied Soft Computing Journal*, vol. 9, no. 2, pp. 658-666, 2009.
- [4] T. Chen, "A fuzzy-neural and multiple-bucket approach for estimating lot cycle time in a wafer fab with dynamic product mix," *Computers and Industrial Engineering*, vol. 55, pp. 423-438, 2008.
- [5] T. Chen, "A hybrid fuzzy-neural approach to job completion time prediction in a semiconductor fabrication factory," *Neurocomputing*, vol. 71, no. 16-18, pp. 3193-3201, 2008.
- [6] 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.
- [7] 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.
- [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] 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.
- [10] T. Chen, "A fuzzy-neural DBD approach for job scheduling in a wafer fabrication factory," *International Journal of Innovative Computing, Information and Control*, vol. 8, no. 6, pp. 4024-4044, 2012.
- [11] S. L. Yang, M. Liu, and L. Li, "Forecasting of production cycle of engineer-to-order products," in *Proceedings of the IEEE 18th International Conference on Industrial Engineering and Engineering Management*, pp. 510-513, 2011.
- [12] T. Chen, "A fuzzy-neural knowledge-based system for job completion time prediction and internal due date assignment in a wafer fabrication plant," *International Journal of Systems Science*, vol. 40, no. 8, pp. 889-902, 2009.
- [13] W. L. Pearn, S. L. 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.
- [14] C. F. Chien, C. Y. Hsu, and C. W. Hsiao, "Manufacturing intelligence to forecast and reduce semiconductor cycle time," *Journal of Intelligent Manufacturing*, vol. 23, no. 6, pp. 2281-2294, 2011.
- [15] 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: Theory Applications and Practice*, vol. 10, no. 1, pp. 55-61, 2003.
- [16] 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.
- [17] T. Chen, “An intelligent hybrid system for wafer lot output time prediction,” *Advanced Engineering Informatics*, vol. 21, no. 1, pp. 55–65, 2007.
- [18] 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.
- [19] T. Beeg, “Wafer fab cycle time forecast under changing loading situations,” in *Proceedings of the IEEE Conference and Workshop on Advanced Semiconductor Manufacturing*, pp. 339–343, May 2004.
- [20] 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.
- [21] 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.
- [22] 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.
- [23] 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.
- [24] T. Chen, “Incorporating fuzzy c -means and a back-propagation network ensemble to job completion time prediction in a semiconductor fabrication factory,” *Fuzzy Sets and Systems*, vol. 158, no. 19, pp. 2153–2168, 2007.
- [25] 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.
- [26] 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. 9–12, pp. 1007–1018, 2011.
- [27] T. Chen, “Evaluating the mid-term competitiveness of a product in a semiconductor fabrication factory with a systematic procedure,” *Computers and Industrial Engineering*, vol. 53, no. 3, pp. 499–513, 2007.
- [28] T. Chen, “A PCA-FBPN approach for job cycle time estimation in a wafer fabrication factory,” *International Journal of Fuzzy System Applications*, vol. 2, no. 2, pp. 50–67, 2012.
- [29] X. He and Q. He, “Application of PCA method and FCM clustering to the fault diagnosis of excavator’s hydraulic system,” in *Proceedings of the IEEE International Conference on Automation and Logistics (ICAL ’07)*, pp. 1635–1639, August 2007.
- [30] 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.
- [31] H.-C. Chang, G.-S. Liang, C.-W. Chu, and C.-H. Chou, “Prioritizing service attributes for improvement using fuzzy zone of tolerance,” *International Journal of Innovative Computing, Information and Control*, vol. 8, no. 1, pp. 75–89, 2012.
- [32] X. Su, P. Shi, L. Wu, and Y. D. Song, “A novel control design on discrete-time Takagi-Sugeno fuzzy systems with time-varying delays,” *IEEE Transactions on Fuzzy Systems*. In press.
- [33] X. Su, P. Shi, L. Wu, and Y.-D. Song, “A novel approach to filter design for T-S fuzzy discrete-time systems with time-varying delay,” *IEEE Transactions on Fuzzy Systems*, vol. 20, no. 6, Article ID 6189779, pp. 1114–1129, 2012.
- [34] L. Wu, X. Su, P. Shi, and J. Qiu, “Model approximation for discrete-time state-delay systems in the T-S fuzzy framework,” *IEEE Transactions on Fuzzy Systems*, vol. 19, no. 2, pp. 366–378, 2011.
- [35] L. Wu and W. X. Zheng, “L2-L ∞ control of nonlinear fuzzy it δ stochastic delay systems via dynamic output feedback,” *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, vol. 39, no. 5, pp. 1308–1315, 2009.
- [36] T. Niknam, H. D. Mojarrad, and M. Nayeripour, “A new hybrid fuzzy adaptive particle swarm optimization for non-convex economic dispatch,” *International Journal of Innovative Computing, Information and Control*, vol. 7, no. 1, pp. 189–202, 2011.
- [37] S. Y. Cho, C. W. Ting, and C. Quek, “Thermal facial pattern recognition for personal verification using fuzzy cmac model,” *International Journal of Innovative Computing, Information and Control*, vol. 7, no. 1, pp. 203–222, 2011.
- [38] R. Yang, Z. Zhang, and P. Shi, “Exponential stability on stochastic neural networks with discrete interval and distributed delays,” *IEEE Transactions on Neural Networks*, vol. 21, no. 1, pp. 169–175, 2010.
- [39] R. Yang, H. Gao, and P. Shi, “Novel robust stability criteria for stochastic Hopfield neural networks with time delays,” *IEEE Transactions on Systems, Man, and Cybernetics B*, vol. 39, no. 11, pp. 467–474, 2009.
- [40] T. Chen and Y. C. Wang, “A fuzzy-neural system with error feedback to adjust classification for forecasting wafer lot flow time: a simulation study,” *Proceedings of the Institution of Mechanical Engineers. Part I: Journal of Systems and Control Engineering*, vol. 221, no. 5, pp. 807–817, 2007.
- [41] T. Chen, “A collaborative fuzzy-neural system for global CO₂ concentration forecasting,” *International Journal of Innovative Computing, Information and Control*, vol. 8, no. 11, pp. 7679–7696, 2012.
- [42] F. Wilcoxon, “Individual comparisons by ranking methods,” *Biometrics Bulletin*, vol. 1, no. 6, pp. 80–83, 1945.



Hindawi

Submit your manuscripts at
<http://www.hindawi.com>

