Int J Adv Manuf Technol (2008) 37:782–792
DOI 10.1007/s00170-007-1007-y

ORIGINAL PAPER

A SOM-FBPN-ensemble approach with error feedback to adjust classification for wafer-lot completion time prediction

Toly Chen

Received: 13 November 2006 / Accepted: 12 March 2007 / Published online: 28 March 2007

Abstract Predicting the completion time of a lot is a critical task to a wafer fabrication plant (wafer fab). Many recent studies have shown that pre-classifying a wafer lot before predicting the completion time was beneficial to prediction accuracy. However, most classification approaches applied in this field could not absolutely classify wafer lots. Besides, whether the pre-classification approach combined with the subsequent prediction approach was suitable for the data was questionable. For tackling these two problems, a self-organization map-fuzzy-back-propagation network-ensemble (SOM-FBPN-ensemble) approach with error feedback to adjust classification is proposed in this study. The proposed methodology has two advanced features: predicting the completion time using a FBPN-ensemble instead of a single FBPN, and feeding back the prediction error to adjust the classification result by the SOM. According to experimental results, the prediction accuracy of the proposed approach was significantly better than those of many existing approaches. Besides, the effects of the two advanced features were also evident.

Keywords Completion time prediction · Ensemble · Error feedback · Fuzzy back propagation network · Self-organization map · Wafer fab

T. Chen (✉)
Department of Industrial Engineering and Systems Management,
Feng Chia University,
No. 100, Wenhwa Rd., Seatwen,
Taichung City, Taiwan
e-mail: tolychen@ms37.hinet.net

1 Introduction

Predicting the completion time for every lot in a wafer fabrication plant (wafer fab) is a critical task not only to the fab itself, but also to its customers. After the completion time of each lot in a wafer fab is accurately predicted, several managerial goals (including internal due-date assignment, output projection, ordering decision support, enhancing customer relationship, and guiding subsequent operations) can be simultaneously achieved [8]. Predicting the completion time of a lot is equivalent to estimating the cycle (flow) time of the lot, because the former can be easily derived by adding the release time (a constant) to the latter.

There are six major approaches commonly applied to predict the completion/cycle time of a wafer lot: multiple-factor linear combination (MFLC), production simulation (PS), back propagation networks (BPN), case-based reasoning (CBR), fuzzy modeling methods, and hybrid approaches [11]. Among the six approaches, MFLC is the easiest, quickest, and most prevalent in practical applications. The major disadvantage of MFLC is the lack of forecasting accuracy [8]. Conversely, a huge amount of data and lengthy simulation time are two disadvantages of PS. Nevertheless, PS is the most accurate completion-time prediction approach if the related databases are continuously updated to maintain enough validity, and it often serves as a benchmark for evaluating the effectiveness (prediction accuracy) of another method. Considering both effectiveness and efficiency (execution time), Chang et al. [5], Chang and Hsieh [3], and Hsu and Sha [14] all forecasted the completion time of a lot in a wafer fab with a BPN having a single hidden layer. Compared with MFLC approaches, the average prediction accuracy measured with root mean squared error (RMSE) was considerably improved with these BPNs. For example, an improvement of about

40% in RMSE was achieved in Chang et al. [5]. On the other hand, much less time and fewer data are required to generate a completion-time forecast with a BPN than with PS. More recently, Chen [11] incorporated the future release plan of the fab into a BPN, and constructed a "look-ahead" BPN for the same purpose, which led to an average reduction of 12% in RMSE. Chang et al. [4] proposed a *k*-nearest-neighbors based case-based reasoning (CBR) approach which outperformed the BPN approach in forecasting accuracy. In one case, the advantage of CBR over BPN was up to 27%. Other CBR applications refer to [2, 7]. Chang et al. [5, 7] 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 [16], called the WM method, with a simple genetic algorithm (GA) and proposed the evolving fuzzy rule (EFR) approach to predict the cycle time of a lot in a wafer fab. Their EFR approach outperformed CBR and BPN in prediction accuracy. Chen [8] constructed a FBPN that incorporated expert opinions in forming inputs to the FBPN. Chen's FBPN was a hybrid approach (fuzzy modeling and BPN) and surpassed the crisp BPN especially in respect to efficiency. Another hybrid approach was proposed in Chang and Liao [6] by combining SOM and WM, in which a lot was classified using a SOM before predicting the completion time of the lot with WM. Chen [12] 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 [9] proposed a look-ahead SOM-FBPN approach for wafer-lot completion time prediction. Besides, a set of fuzzy inference rules were also established to evaluate the achievability of an estimated completion time. Subsequently, Chen et al. [13] added a selective allowance to the completion time predicted using Chen's approach to determine the internal due date.

According to these results, classifying wafer lots is a good way of getting better performance in lot completion time prediction. However, most classification approaches applied in this field could not absolutely classify wafer lots. In other words, a wafer lot can be classified into multiple categories with different degrees. Therefore, predicting the completion time of a wafer lot using the prediction approach tailored to a single category seems to be insufficient. On the other hand, whether the pre-classification approach combined with the subsequent prediction approach is suitable for the data is questionable. For tackling these two problems, Chen's look-ahead SOM-FBPN approach is modified and the look-ahead SOM-FBPN-ensemble approach with error feedback to adjust classification is proposed in this study. In the proposed methodology,

(1) A wafer lot is classified using a SOM before predicting the completion time of the wafer lot with a FBPN-ensemble.
(2) "Many" FBPNs form a FBPN ensemble that is applied to wafer lot completion time prediction. For this purpose, an output aggregation mechanism is also proposed. Conversely, only "a single" network or rule base is applied in existing approaches. This opens a new direction for future research in this field, because the other approaches can be improved in the same way to enhance the effectiveness.
(3) Many pre-classification techniques (e.g., kM, FCM, SOM) have been applied in this field. However, whether the pre-classification techniques combined with the subsequent prediction approach (e.g., kM-FBPN, FCM-FBPN, SOM-FBPN, SOM-WM, etc.) is suitable for the data is questionable, which is very critical to the performance. Instead of trying many classifier-and-predictor combinations, the concept of feedback control is applied in this study to improve the suitability of the SOM-FBPN combination for the data by feeding back the forecasting error generated by the FBPN-ensemble to adjust the classification result done by the SOM. After some replications, the SOM-FBPN combination will become more suitable for the data. Such a treatment is also novel in this field, and opens another direction for future research, because the other pre-classifying approaches can also be tuned in similar ways.

The methodology architecture is shown in Fig. 1. The remaining of this paper is organized as follows. Section 2 introduces the SOM-FBPN-ensemble approach with error feedback to adjust classification. To evaluate the effectiveness of the proposed methodology, PS is applied in Sect. 3 to generate test data. Based on analysis results, some discussions are made in Sect. 4. Finally, the concluding remarks and some directions for future research are given in Sect. 5.

2 A SOM-FBPN-ensemble with error feedback to adjust classification

The proposed methodology is modified from Chen's look-ahead SOM-FBPN approach [9]. Parameters that will be used in the proposed methodology are defined:

a_n The normalized cycle time of lot n that is calculated as (the cycle time of lot n - The minimal cycle time) / (the maximal cycle time - the minimal cycle time). There is only one fixed value of a_n no matter which category lot n is re-classified into.

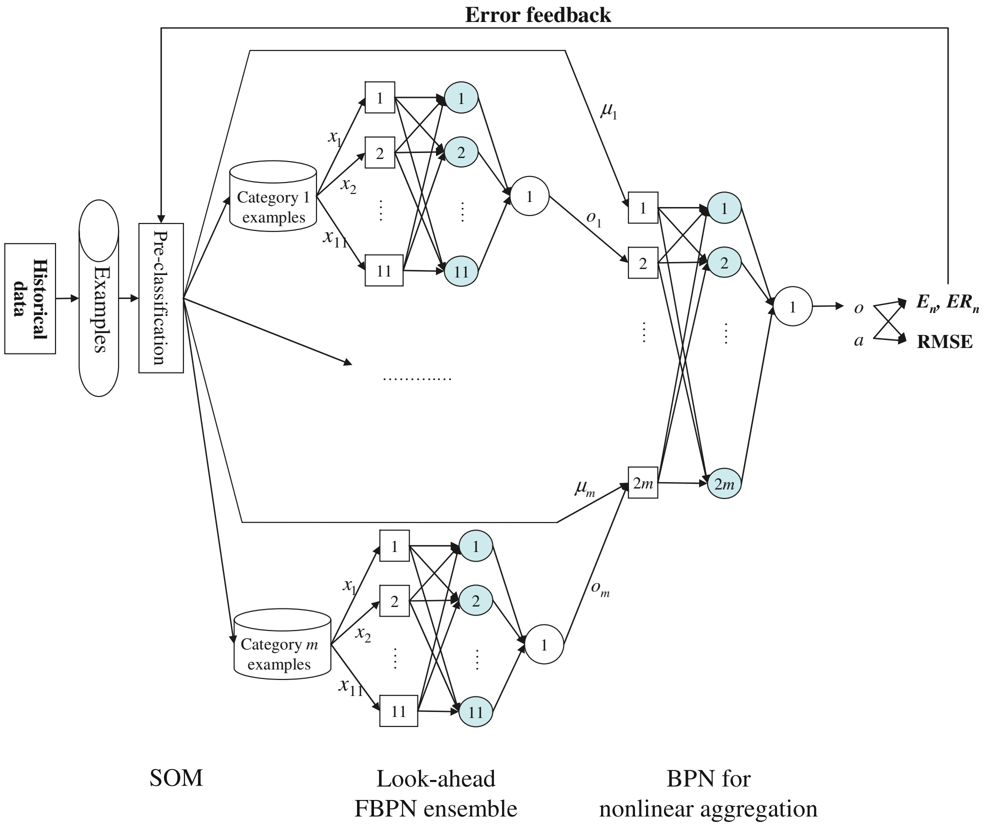


Fig. 1 The methodology architecture

o_n The normalized cycle time forecast of lot n, which changes every training epoch and is expected to converge to a_n.

R_n The release time of lot n.

U_n The average fab utilization at R_n.

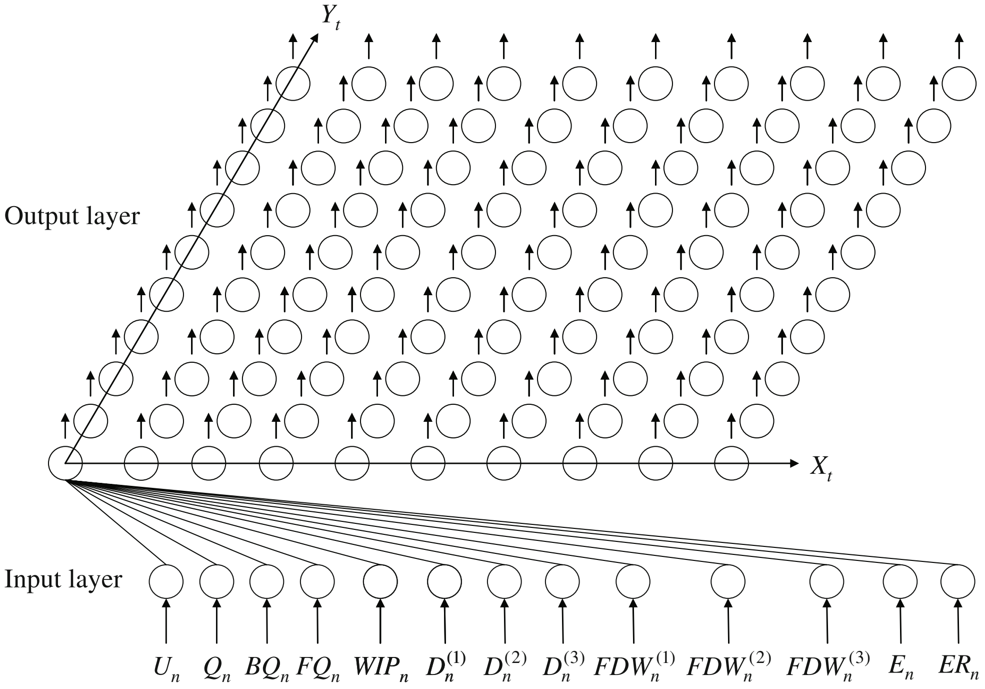
Q_n The total queue length on the processing route of lot n at R_n.

BQ_n The total queue length before bottlenecks at R_n.

FQ_n The total queue length in the whole fab at R_n.

WIP_n The fab work-in-progress (WIP) at R_n.

Fig. 2 The SOM structure



$D_n^{(r)}$ The delay (waiting time) of the rth recently completed lot at R_n; r = 1~3.
$FDW_n^{(f)}$ The future discounted workload on the processing route of lot n at R_n [9]; f=1~3.
E_n The prediction error of lot n.
ER_n The prediction error rate of lot n.

There are three steps in applying the proposed methodology to predict the completion time of a wafer lot. The first step is to pre-classify wafer lots with a SOM, which has been shown to be effective in improving the accuracy of wafer lot completion time prediction [6, 9, 10, 12, 13].

2.1 Step 1: Wafer lot pre-classification with a SOM

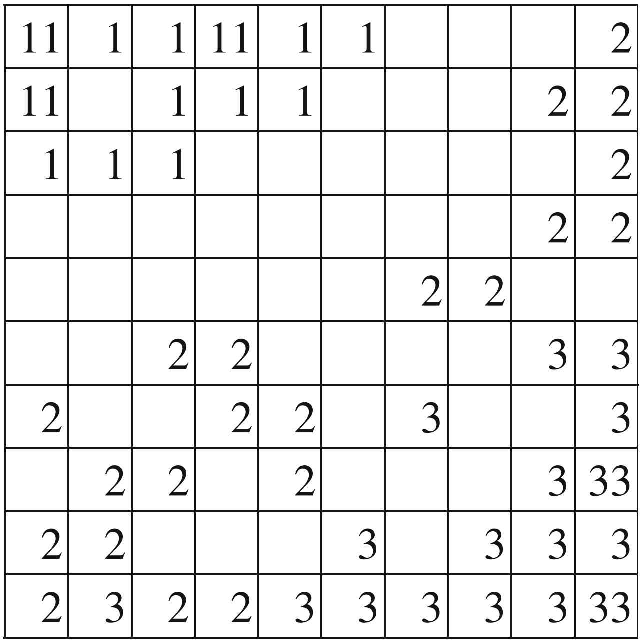
Every lot fed into the FBPN ensemble is called an example. Examples are pre-classified into different categories with a SOM. A SOM is usually trained using unsupervised learning to produce low dimensional representation of the training examples while preserving the topological properties of the input space. For this reason, the two dimensional SOM constructed in this study ought to be able to manage the 13 dimensional data of wafer lots. In addition, there have been some studies using SOM for the same purpose, e.g., Chang and Liao [6] and Chen [9, 10, 13]. The structure of the SOM is 10*10, and the number of output nodes is 100. Let x_n denote the 13-dimensional feature vector (U_n, Q_n, BQ_n, FQ_n, WIP_n, $D_n^{(1)}$, $D_n^{(2)}$, $D_n^{(3)}$, $FDW_n^{(1)}$, $FDW_n^{(2)}$, $FDW_n^{(3)}$, E_n, ER_n) corresponding to lot n (see Fig. 2). Note that the prediction error E_n and the prediction error rate ER_n are included, and they are set to be zeros in the beginning. The feature vectors of all lots are fed into the SOM to be learned. The learning algorithm is the same as that adopted in Chen [9]. However, pre-classification was done only once in Chen's study, while lots are pre-classified many times in the proposed methodology. After that, a labeling process is realized. According to the distribution of labeled (categorized) output nodes of the SOM after the labeling process, merged or isolated clusters can be visually analyzed, and the number of categories can be clarified. A case is illustrated in Fig. 3. According to this figure, examples can be classified into three categories. The centroid of each category is calculated as:

$$\bar{x}_{(k)} = \left\{\bar{x}_{(k)i} \middle| i = 1 \sim 13\right\},$$

$$\bar{x}_{(k)i} = \sum_{all\,v} x_{vi} \Bigg/ \sum_{all\,v} y_{vi},$$

$$y_{vi} = \left\{ {}^{1}_{0}\; if \; {}^{example\,v \in category\,k,}_{else,} \right.$$

$$k = 1 \sim 3.$$



11	1	1	11	1	1				2
11		1	1	1				2	2
1	1	1							2
								2	2
						2	2		
		2	2					3	3
2			2	2		3			3
	2	2		2				3	33
2	2				3		3	3	3
2	3	2	2	3	3	3	3	3	33

Fig. 3 Labeling of the SOM (product B, hot lots)

where $\bar{x}_{(k)}$ denotes the centroid of category k. However, it is often difficult to classify an example into a single category absolutely. Therefore, the membership of an example belonging to each category is calculated:

$$\mu_{v(k)} = 1 \Bigg/ \sum_{l=1}^{3} \left(e_{v(k)}/e_{v(l)}\right)^2,$$

$$e_{v(k)} = \sqrt{\sum_{\text{all } i} \left(x_{vi} - \bar{x}_{(k)i}\right)^2},$$

where $\mu_{v(k)}$ denotes the membership of example v belonging to category k. In this way, an example can be classified into multiple categories with different degrees. Conversely, in Chang and Liao [6] and Chen [9] an example is only classified into a single category (the category with the highest membership) to simplify the situation.

After classification, examples of different categories are then learned with different FBPNs but with the same topology. Before that, a membership threshold μ_L for adopting an example in network learning has to be determined. Only examples which membership values of belonging to a category are greater than or equal to μ_L will be adopted in training the FBPN to obtain the parameter values tailored to the category. As a result, an example might be adopted by multiple categories. The procedure for determining the parameter values of FBPNs is described in the next section.

2.2 Step 2: Completion time prediction within each lot category with FBPN

The configuration of the FBPN is established as follows:

(1) Inputs: eleven parameters associated with the n-th example/lot including U$_n$, Q$_n$, BQ$_n$, FQ$_n$, WIP$_n$, $D_n^{(r)}$ (r=1~3), and $FDW_n^{(f)}$ (f=1~3). These parameters have to be normalized so that their values fall within [0, 1]. Then some production execution/control experts are requested to express their beliefs (in linguistic terms) about the importance of each input parameter in predicting the cycle/completion time of a lot. Linguistic assessments for an input parameter are converted into several pre-specified fuzzy numbers. The subjective importance of an input parameter is then obtained by averaging the corresponding fuzzy numbers of the linguistic replies for the input parameter by all experts. The subjective importance obtained for an input parameter is multiplied to the normalized value of the input parameter. After such a treatment, all inputs to the FBPN become triangular fuzzy numbers (TFNs), and the fuzzy arithmetic for TFNs is applied to deal with all calculations involved in training the FBPN.

(2) Single hidden layer: Generally one or two hidden layers are more beneficial for the convergence property of the FBPN.

(3) Number of neurons in the hidden layer: the same as that in the input layer. Such a treatment has been adopted by many studies (e.g., [4, 8–11]).

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

(5) Network learning rule: delta rule.

(6) Transformation function: Sigmoid function,

$$f(x) = \frac{1}{1 + e^{-x}}.$$

(7) Learning rate (η): 0.01~1.0.

(8) Batch learning.

(9) Number of epochs per replication: 75000.

(10) Number of initial conditions/replications: 100. Because the performance of a BPN or FBPN is sensitive to the initial condition, the training or testing process will be repeated many times with different initial conditions that are randomly generated. Among the results, the best one is chosen for the subsequent analyses.

The procedure for determining the parameter values refers to Chen [9], and is only briefly described here. 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 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 are outputted from the hidden layer, and also transferred to the output layer with the same procedure. Finally, the output of the FBPN $\tilde{o}_n$ is generated. $\tilde{o}_n$ is defuzzified according to the centroid-of-area (COA) formula, and then the defuzzification result o_n is compared with the actual value (the normalized cycle time) a_n to evaluate the accuracy of the FBPN which is represented with RMSE. In addition, the prediction error E_n and the prediction error rate ER_n can be calculated as:

$$E_n = o_n - a_n,$$
$$ER_n = E_n / a_n,$$

The two parameters will be fed back to the SOM, and classification is done again. The rationale for such a treatment is explained as follows. In the proposed way, lots that are topologically close to each other and have similar values of the two parameters will be re-classified into the same category, which is beneficial to improving the prediction accuracy of the FBPN. Take the extreme case in which all lots of a category have the same values of E_n as an example. Namely, $E_{(p)} = o_{(p)} - a_{(p)} = E \; \forall$ lot p belonging to the category.

Simply add -E (a constant) to the network output of the FBPN, and the prediction accuracy of all lots in the category can be simultaneously enhanced: *new* $E_{(p)} =$ *new* $o_{(p)} - a_{(p)} = o_{(p)} - E - a_{(p)} = o_{(p)} - (o(p) - a(p)) - a(p) = 0, \; \forall$ lot p belonging to the category.

Incorporating the effect of adding a constant through a few epochs of re-training ought to be easy, because theoretically a well-trained BPN or FBPN (without being stuck to local minima) with a good selected topology can successfully map any complex nonlinear distribution.

Subsequently in the backward phase, the deviation between o_n and a_n is propagated backward, and the error terms of neurons in the output and hidden layers can be calculated. Based on them, adjustments that should be made to the connection weights and thresholds can be obtained. To accelerate convergence, a momentum can be added to the learning expressions as well. Network-learning stops when RMSE falls below a pre-specified level, or the improvement in RMSE becomes negligible with more epochs, or a large number of epochs have already been run. Besides, the lower and upper bounds of all fuzzy numbers in the FBPN will no longer be modified if Chen's index [9] converges to a minimal value. Then the remaining portion of the adopted examples in each category is used as "testing examples" and fed into the FBPN to evaluate the accuracy of the network again that is also measured with

RMSE. However, re-classification might lead to the situation that lots keep transferring from one category to another, and the convergence of any FBPN might not be achieved. In this respect, according to experimental data, the results of classification will quickly converge, because among the 13 inputs to the SOM only two of them (E_n and ER_n) are variable and might change the classification results. E_n and ER_n will become smaller and smaller because of network learning, and are expected to converge to zeros. Finally lot transferring will stop, and every FBPN will converge.

After training and testing, the FBPN of every category is applied to predict the cycle times of examples belonging to the category. Then the prediction error E_n and the prediction error rate ER_n are calculated for every example, and are fed back to the SOM classifier. Subsequently, classification is done again. If the classification result is the same as the previous one, then stop; otherwise, the FBPN of every category has to be re-trained and re-tested again with the same procedure mentioned above. Note that only the examples input to the FBPN might be different. The structure of the FBPN itself is not changed. Finally, these FBPNs form a FBPN-ensemble that can be applied to predict the cycle/completion time of a new lot. The procedure is detailed in the next section.

2.3 Step 3: Aggregation of outputs from the component FBPNs

To obtain a better prediction result, a BPN is constructed to (nonlinearly) aggregate the outputs from the component FBPNs of the FBPN-ensemble with the following configuration:

(1) Inputs: 2 m parameters including the outputs (indicated with $o_{v(k)}$; k=1~m) from the component FBPNs obtained for lot v, and the membership value of lot v belonging to each component FBPN (indicated with $\mu_{v(k)}$).
(2) Single hidden layer.
(3) Number of neurons in the hidden layer: the same as that in the input layer.
(4) Output: the (normalized) cycle time forecast of the lot.
(5) Network learning rule: Delta rule.
(6) Transformation function: Sigmoid function.
(7) Learning rate (η): 0.01~1.0.
(8) Batch learning.
(9) Number of epochs per replication: 75000.
(10) Number of initial conditions/replications: 100.

The BPN also undergoes training and testing. Then, it is applied and the network output (i.e., the aggregation result) determines the cycle time forecast for the new lot. In this study, SOM and BPN were both implemented on the software "NeuroSolutions 4.0", while a VB program has been constructed to implement FBPN.

3 Test data from a simulated wafer fab

In real situations, the historical data of each lot is only partially available in the fab. Further, some information of the previous lots such as Q_n, BQ_n, and FQ_n is not easy to obtain on the shop floor. Therefore, a simulation model is often built for the manufacturing process of a real wafer fab [1, 3–6, 8–11, 14, 15]. Then, such information can be derived from the shop floor status collected from the simulation model [4]. To generate some test examples, a simulation program coded using Microsoft Visual Basic 6.0 is constructed to simulate a wafer fabrication environment with the following assumptions:

(1) The distributions of the times between the adjacent machine breakdowns are exponential.
(2) The distribution of the time required to repair a machine is uniform.
(3) The percentages of lots with different product types in the fab are predetermined. As a result, this study is only focused on fixed-product-mix cases.
(4) The percentages of lots with different priorities released into the fab are controlled.
(5) Lots are sequenced on each machine first by their priorities, then by the first-in-first-out (FIFO) policy. Such a sequencing policy is not uncommon in many foundry fabs.
(6) A lot has equal chances to be processed on each alternative machine/head available at a step.
(7) A lot cannot proceed to the next step until the fabrication on its every wafer has been finished.
(8) No preemption is allowed.

The basic configuration of the simulated wafer fab is the same as a real-world wafer fab which is located in the Science Park of Hsin-Chu, Taiwan, R.O.C. Assumptions (1)~(2), and (5)~(8) are commonly adopted in related researches (e.g., [3–6, 8]), while assumptions (3)~(4) are made to simplify the situation. There are five products (labeled as A~E) in the simulated fab. A fixed product mix is assumed. The percentages of these products in the fab's product mix are assumed to be 35%, 24%, 17%, 15%, and 9%, respectively. The simulated fab has a monthly capacity of 20,000 pieces of wafers and is expected to be fully utilized (utilization=100%). Lots are uniformly (every a fixed interval) released into the fab, and have a standard size of 24 wafers per lot. The mean inter-release time of lots into the fab is 0.88 h. Three types of priorities (normal, hot, and super hot) are randomly assigned to lots. The percentages of lots with these priorities released into the fab are restricted to be approximately 60%, 30%, and 10%, respectively. Each product has 150~200 steps and 6~9 reentrances to the most bottleneck machine. The singular production characteristic “reentry” of the semiconductor industry is clearly reflected in the simulation model. It also shows the difficulty for the production planning and scheduling staff to provide an accurate due-date for the product with such a complicated routing. Totally 102 machines (including alternative machines) are used to process single-wafer or batch operations in the fab. Thirty replications of the simulation are successively run. The time required for each simulation replication is about 15 minute on a PC with 256MB RAM and Athlon 64 Processor 3000+ CPU. A horizon of 24 months is simulated. The maximal cycle time is less than 3 months. Therefore, 4 months and an initial WIP status (obtained from a pilot simulation run) seemed to be sufficient to drive the simulation into a steady state. The statistical data were collected starting at the end of the fourth month. For each replication, data of 30 lots are collected and classified by their product types and priorities. In total, data of 900 lots can be collected as training and testing examples. Among them, 2/3 (600 lots, including all product types and priorities) are used to train the network, and the other 1/3 (300 lots) are reserved for testing. The three parameters in calculating the future discounted workloads are specified as: T_1=one week; T_2= 1.5 weeks; T_3=2 weeks.

A traced report was generated every simulation run for verifying the simulation model. The average cycle times have also been compared with the actual values for validating the simulation model.

4 Results and discussions

To evaluate the effectiveness of the proposed methodology and to make comparison with some existing approaches - BPN, FBPN, CBR, EFR, kM-FBPN, SOM-WM, and the look-ahead SOM-FBPN (without ensemble and error feedback), all the eight methods were applied to five test cases containing the data of full-size (24 wafers per lot) lots with different product types and priorities.

In BPN or FBPN, there was one hidden layer with 11 nodes. In SOM-WM, look-ahead SOM-FBPN, and the proposed methodology, lots were pre-classified with SOM. The number of categories (m) determined for each product type and priority is shown in Table 1. In kM, the optimal number of categories (K) (see Table 2) was obtained by applying Xie and Beni’s S test [17]. The convergence condition in training networks was established as either the improvement in RMSE becomes less than 0.001 with one more epoch, or 75000 epochs have already been run.

Table 1 The number of lot categories determined by SOM

	A (normal)	A (hot)	A (super hot)	B (normal)	B (hot)
m	6	4	4	5	3

The minimal RMSEs achieved by applying the eight approaches to different cases were recorded and compared in Table 3. As noted in Chang et al. [5], the k-nearest-neighbors based CBR approach should be compared with a BPN trained with only randomly chosen k cases. The optimal value of parameter k in the CBR approach was equal to the value that minimized RMSE [5]. The k values for different product types and priorities are summarized in Table 4. BPN was adopted as the comparison basis, and the percentage of improvement on the minimal RMSE by applying another approach is enclosed in parentheses following the performance measure. According to experimental results,

(1) From the effectiveness viewpoint, the prediction accuracy (measured with RMSE) of the proposed approach was significantly better than those of the other approaches by achieving a 28%~55% (and an average of 43%) reduction in RMSE over the comparison basis - the BPN approach. The average advantages over FBPN, CBR, and EFR were 41%, 38%, and 15%, respectively.

(2) The proposed approach surpassed kM-FBPN, SOM-WM, and the look-ahead SOM-FBPN that performed pre-classification as well by improving the prediction accuracy up to 12%, 11%, and 9%, respectively.

(3) The performances of EFR and SOM-WM were very close. In fact, these two approaches are quite similar in nature. Logical rule sets such as WM and EFR classify sample first, and then provide different treatments. Adding another classifier (SOM) to them seems to have little effect.

(4) As the lot priority rose, the superiority of the proposed approach over approaches without pre-classification increased. In fact, the cycle time variation of lots with higher priorities is often smaller, which makes their cycle times easier to predict. Clustering such lots has a more significant effect on the performance of cycle time prediction.

Table 2 The optimal number of lot categories determined by kM

	A (normal)	A (hot)	A (super hot)	B (normal)	B (hot)
K	8	6	3	9	5

Table 3 Comparisons of the RMSEs of various approaches

RMSE	BPN	FBPN	CBR	EFR	kM-FBPN	SOM-WM	SOM-FBPN	The proposed approach
A (normal)	178.59	177.1 (−1%)	172.44(−3%)	164.29 (−8%)	157.78 (−12%)	160.25 (−10%)	141.47 (−21%)	129.32 (−28%)
A (hot)	102.1	102.27 (+0%)	86.66 (−15%)	66.21 (−35%)	64.93 (−36%)	61.51 (−40%)	59.51 (−42%)	48.84 (−52%)
A (super hot)	13.49	12.23 (−9%)	11.59 (−14%)	9.07 (−33%)	9.48 (−30%)	9.07 (−33%)	9.07 (−33%)	8.62 (−36%)
B (normal)	289.22	286.93 (−1%)	295.51 (+2%)	208.28 (−28%)	197.1 (−32%)	198.5 (−31%)	178.42 (−38%)	158.63 (−45%)
B (hot)	77.61	75.98 (−2%)	78.85 (+2%)	44.57 (−43%)	42.01 (−46%)	39.67 (−49%)	38.59 (−50%)	35.21 (−55%)

(5) Conversely, the greatest superiority of the proposed approach over EFR happened when the lot priority was the smallest (normal lots).

To demonstrate the effect of feeding back the prediction error and error rate to adjust classification, an example is given in Fig. 4, in which the fluctuations in the membership values of some lot belonging to three categories, indicated with μ_1~μ_3, before and after a few replications of classification adjustment are compared. According to this figure, the classification result did change after feeding back the prediction error and error rate. Nevertheless, μ_1~μ_3 seemed to converge. The fluctuations in the membership values corresponded to the transitions of the distributions of labeled (categorized) output nodes of SOM, as demonstrated in Fig. 5. To evaluate the effect of feeding back the prediction error and error rate to adjust classification, the prediction accuracy of the look-ahead SOM-FBPN with error feedback was compared with that of the same approach but without error feedback in Table 5. The advantage was 5.4% on average.

Subsequently, to elaborate the performance of the proposed methodology with respect to the algorithm used for aggregation, the common weighted average (WA) was also applied to perform linear aggregation, which resulted in the aggregation-by-WA approach, while the proposed approach was called the aggregation-by-BPN approach. The third approach compared with did not perform aggregation and was therefore called the no-aggregation approach, in which only the FBPN with the highest membership would be applied to predict the cycle time. For a fair comparison, none of the three approaches fed back the prediction error to adjust classification. The forecasting results were summarized in Table 6. The effect of aggregation (using a network ensemble instead of a single network) was revealed with the average advantage of the aggregation-by-BPN approach over the no-aggregation approach, which was 8.2%. Besides, the performance of the aggregation-by-WA approach was even worse than that of the no-aggregation approach, which meant that casual aggregation (even with the common WA) did not lead to a better result.

Table 4 The *k* values for different product types and priorities in CBR

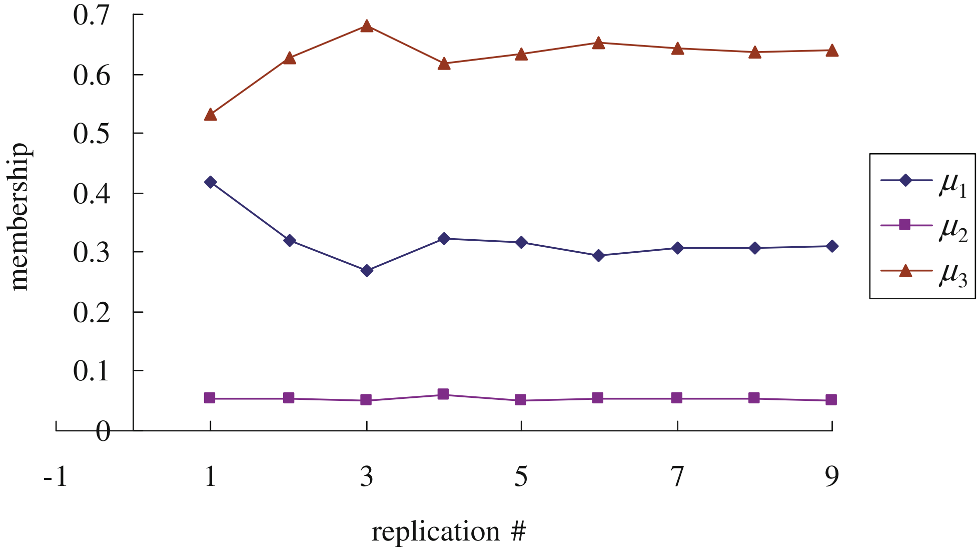
	A (normal)	A (hot)	A (super hot)	B (normal)	B (hot)
k	8	6	4	9	5

On the other hand, the prediction efficiency becomes worse after incorporating in the new characteristics. Feeding back the forecasting error and error rate to adjust the classification results and then re-training and re-testing the FBPNs does prolong the learning time. After that, it also takes a little more time to predict the cycle time of a new lot with a FBPN ensemble than with a single FBPN. For demonstrating this, the eight approaches were implemented using MATLAB R2006a and Microsoft VB.NET on a PC with 256MB RAM and Athlon 64 Processor 3000+ CPU. Two performance measures including the model construction/learning time, and the forecasting time for every product type and priority were collected. The results were summarized in Table 7. Take the data of product type A with normal priority as an example, after incorporating in the new characteristics the learning time was lengthened from 307 minutes to 573 minutes. Secondly, the forecasting times by applying various approaches were compared. The forecasting time by applying PS was lengthy (27189 seconds), while that of the proposed methodology (157 seconds) or any other approach (45~132 seconds) was much shorter. In fact, the cycle time of a wafer lot might be up to 3 months, and therefore forecasting taking only a few minutes is acceptable in practical applications. Besides, effectiveness is usually more emphasized than efficiency. To consider both effectiveness and efficiency, the RMSE and the time forecasting takes for all approaches were compared in Fig. 6, which supported the Pareto optimality of the proposed methodology because it was not dominated by any of the other approaches. Conversely, CBR and FBPN were dominated by EFR. Besides, SOM-WM was dominated by kM-FBPN and look-ahead SOM-FBPN.

5 Conclusions and directions for future research

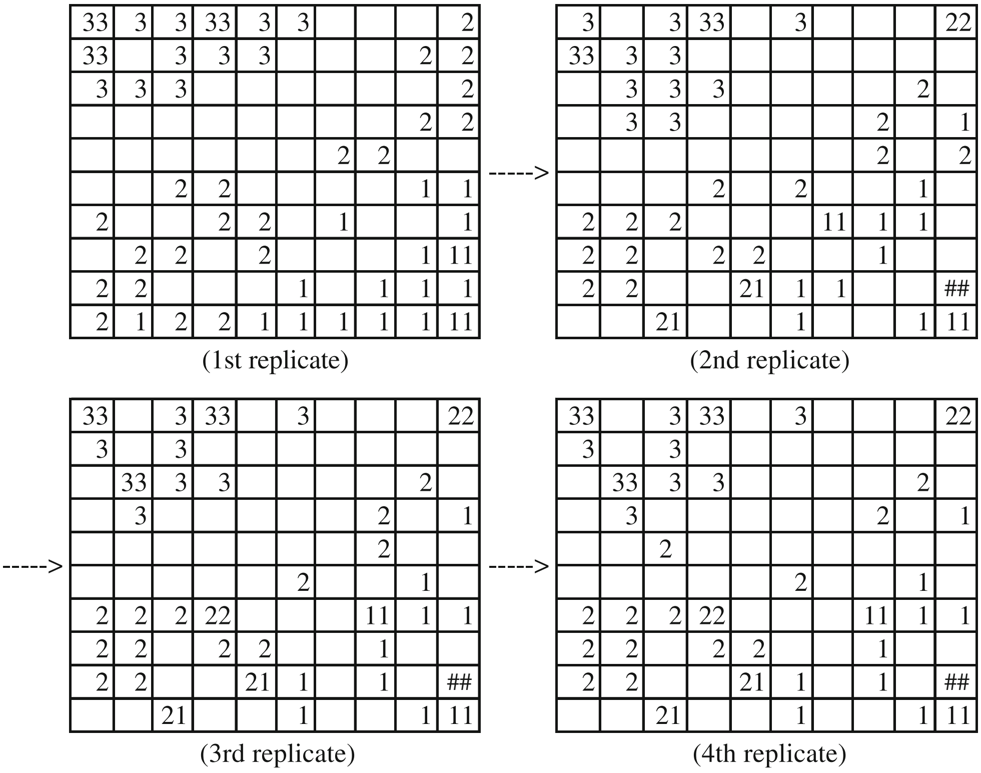
To further enhance the effectiveness of wafer lot completion time prediction, a SOM-FBPN-ensemble approach

Fig. 4 The fluctuations in the membership values



with error feedback to adjust classification is proposed in this study. In the proposed approach, at first wafer lots are pre-classified into several categories using a SOM. For the wafer lots completely belonging to each category, a corresponding look-ahead FBPN is constructed to predict the cycle/completion times. However, most wafer lots belong to multiple categories with different degrees. For this reason, instead of using the FBPN of a single category to predict the cycle time of a new lot, the FBPNs of all categories form a FBPN ensemble that is applied to predict the cycle time. The output of the FBPN ensemble is derived by aggregating the outputs from the component FBPNs with a BPN for nonlinear aggregation. On the other hand, the concept of feedback control is applied to improve the suitability of the SOM-FBPN combination for the data by feeding back the prediction error to adjust the classification result. After some replications, the SOM-FBPN combination will become more suitable for the data. For evaluating the effectiveness of the proposed methodology and to make comparison with some existing approaches, production

Fig. 5 The transitions of the distributions of categorized output nodes of SOM caused by re-classification



33	3	3	33	3	3				2
33		3	3	3				2	2
3	3	3							2
								2	2
						2	2		
		2	2					1	1
2			2	2		1			1
	2	2		2				1	11
2	2				1		1	1	1
2	1	2	2	1	1	1	1	1	11

(1st replicate)

3		3	33		3				22
33	3	3							
	3	3	3					2	
	3	3					2		1
							2		2
			2		2			1	
2	2	2				11	1	1	
2	2		2	2			1		
2	2			21	1	1			##
		21			1			1	11

(2nd replicate)

33		3	33		3				22
3		3							
	33	3	3					2	
	3						2		1
							2		
					2			1	
2	2	2	22				11	1	1
2	2		2	2			1		
2	2			21	1		1		##
		21			1			1	11

(3rd replicate)

33		3	33		3				22
3		3							
	33	3	3					2	
	3						2		1
		2							
					2			1	
2	2	2	22				11	1	1
2	2		2	2			1		
2	2			21	1		1		##
		21			1			1	11

(4th replicate)

Table 5 The effect of feeding back the prediction error and error rate to adjust classification

RMSE	A (normal)	A (hot)	A (super hot)	B (normal)	B (hot)
Without error feedback	141.47	59.51	9.07	178.42	38.59
With error feedback	131.02 (−7%)	54.18 (−9%)	9.05 (−0%)	161.58 (−9%)	37.76 (−2%)

Table 7 The forecasting times by applying various approaches (product A, normal priority)

Approach	Software	Construction/ learning time (min)	Forecasting time (s)
PS	VB program	Several weeks	27189
BPN	MATLAB	195	45
FBPN	VB program	290	49
CBR	VB program	31	47
EFR	VB program	98	46
kM-FBPN	MATLAB + VB program	301	119
SOM-WM	MATLAB + VB program	185	137
Look-ahead SOM-FBPN	MATLAB + VB program	307	132
The proposed approach	MATLAB + VB program	573	157

simulation is applied in this study to generate test data. According to experimental results,

(1) From the effectiveness viewpoint, the prediction accuracy of the proposed methodology was significantly better than those of many existing approaches.
(2) The effect of using a network ensemble instead of a single network for wafer lot completion time prediction is evident.
(3) Feeding back the prediction error did adjust the classification result. In this way, the SOM-FBPN mechanism was tuned to be more suitable for the data, which also contributed to the effectiveness of the proposed methodology.

Conversely, there are disadvantages associated with the proposed methodology:

(1) Compared with some existing approaches, more data are required with the proposed methodology for the sake of classifying lots with a SOM and incorporating the future release plan.
(2) The prediction efficiency becomes worse after incorporating in the two new characteristics.

The main contribution to the body of the knowledge is:

(1) Classifying lots has been shown to be a good way of getting better performance in predicting the completion time of a wafer lot. However, most classification approaches applied in this field could not absolutely classify wafer lots. Therefore, predicting the completion time of a wafer lot using the prediction approach tailored to a single category seems to be insufficient. Instead, aggregating the outputs from the prediction approaches of multiple categories to generate a representative value might be more accurate.
(2) On the other hand, many pre-classification approaches have been applied in this field, e.g., kM, FCM, SOM, etc. However, whether the pre-classification approach combined with the subsequent prediction approach is suitable for the data is questionable. Instead of trying many classification and prediction approaches to find out the most suitable combination, applying the concept of feedback control to improve the suitability of the classification-prediction combination might be more practical.

However, to further evaluate the advantages and disadvantages of the proposed methodology, applying it to cases with changing product mixes or loosely controlled priority combinations is necessary, under which cycle time variation is often very large. These constitute some directions for future research.

Table 6 The effect of aggregation

RMSE	A (normal)	A (hot)	A (super hot)	B (normal)	B (hot)
No-aggregation	141.47	59.51	9.07	178.42	38.59
Aggregation-by-WA	146.78 (+4%)	60.79 (+2%)	9.69 (+7%)	197.10 (+10%)	41.22 (+7%)
Aggregation-by-BPN	132.55 (−6%)	49.94 (−16%)	8.79 (−3%)	162.6 (−9%)	36.01 (−7%)

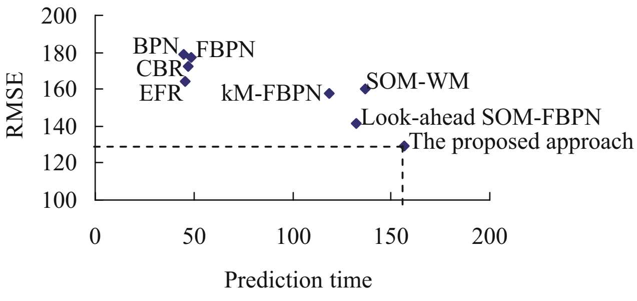


Fig. 6 The Pareto optimality of the proposed methodology

References

1. Barman S (1998) The impact of priority rule combinations on lateness and tardiness. IIE Trans 30:495–504
2. Chang P-C, Chen LY (2006) A hybrid regulation system by evolving CBR with GA for a twin laser measuring system. Int J Adv Manuf Technol, DOI 10.1007/s00170-005-0286-4
3. Chang P-C, Hsieh J-C (2003) A neural networks approach for due-date assignment in a wafer fabrication factory. Int J Ind Eng 10(1):55–61
4. Chang P-C, Hsieh J-C, Liao TW (2001) A case-based reasoning approach for due date assignment in a wafer fabrication factory. Proc International Conference on Case-Based Reasoning (ICCBR 2001), Vancouver, British Columbia, Canada
5. Chang P-C, Hsieh J-C, Liao TW (2005) Evolving fuzzy rules for due-date assignment problem in semiconductor manufacturing factory. J Intell Manuf 16:549–557
6. Chang P-C, Liao TW (2006) Combining SOM and fuzzy rule base for flow time prediction in semiconductor manufacturing factory. Appl Soft Comput 6:198–206
7. Chang P-C, Liu CH, Wang YW (2006) A hybrid model by clustering and evolving fuzzy rules for sale forecasting in printed circuit board industry. Decis Support Syst 42(3):1254–1269
8. Chen T (2003) A fuzzy back propagation network for output time prediction in a wafer fab. Applied Soft Comput 2/3F:211–222
9. Chen T (in press) A hybrid look-ahead SOM-FBPN and FIR system for wafer lot output time prediction and achievability evaluation. Int J Adv Manuf Technol
10. Chen T (2006) A hybrid SOM-BPN approach to lot output time prediction in a wafer fab. Neural Process Lett 24(3):271–288
11. Chen T (2006) A look-ahead fuzzy back propagation network for lot output time series prediction in a wafer fab. Lect Notes Comput Sci 4234:974–982
12. Chen T (2006) An intelligent hybrid system for wafer lot output time prediction. Adv Eng Inform 21:55–65
13. Chen T, Jeang A, Wang Y-C (in press) A hybrid neural network and selective allowance approach for internal due date assignment in a wafer fabrication plant. Int J Adv Manuf Technol
14. Hsu SY, Sha DY (2004) Due date assignment in wafer fabrication using artificial neural network. Int J Adv Manuf Technol 23(9–10):768–775
15. Tiwari MK, Roy D (2002) Minimization of internal shrinkage in casting using synthesis of neural network. Int J Smart Eng Syst Des 4:205–214
16. Wang L-X, Mendel JM (1992) Generating fuzzy rules by learning from examples. IEEE Trans Syst Man Cybern 22(6):1414–1427
17. Xie XL, Beni G (1991) A validity measure for fuzzy clustering. IEEE Trans Patt Anal Mach Intell 13:841–847