Incorporating Future Release Plan in Predicting Wafer Lot Output Time with a Hybrid ANN

TOLY CHEN [a], YU-CHENG LIN [b*] and HSIN-CHIEH WU [c]
[a]Department of Industrial Engineering and Systems Management, Feng Chia University
[b]Department of Industrial Engineering and Management,
The Overseas Chinese Institute of Technology
No. 100, Chiaokwang Road, Taichung City, Taiwan, R.O.C.
[c]Department of Industrial Engineering and Management, Chaoyang University of Technology
*

Abstract: - Output time prediction is a critical task to a wafer fab (fabrication plant). However, traditional wafer lot output time prediction methods are based on the historical data of the fab. The influence of the future release plan has been neglected. In addition, a lot that will be released in the future might appear in front of another lot that currently exists in the fab. For these reasons, to further improve the accuracy of wafer lot output time prediction, the future release plan of the fab has to be considered, and a hybrid ANN (SOM+FBPN) incorporating the future release plan of the fab is proposed in this study. Production simulation is also applied to generate test examples. According to experimental results, the prediction accuracy of the proposed methodology was significantly better than those of three approaches, FBPN, evolving fuzzy rules (EFR), and the hybrid ANN without considering the future release plan in most cases by achieving a 20%~49% (and an average of 35%) reduction in the root-mean-squared-error (RMSE) over the comparison basis – the FBPN.

Key-Words: - Output time prediction; Future release plan; Fuzzy back propagation network; Self-organization map; Wafer fab

1 Introduction

Predicting the output time for every lot in a wafer fab is a critical task not only to the fab itself, but also to its customers. After the output time of each lot in a wafer fab is accurately predicted, several managerial goals can be simultaneously achieved [5]. Predicting the output time of a wafer 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 predicting the output/cycle time of a wafer lot: multiple-factor linear combination (MFLC), production simulation (PS), back propagation networks (BPN), case based reasoning (CBR), fuzzy modelling methods, and hybrid approaches. 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 [5]. Conversely, huge amount of data and lengthy simulation time are two shortages of PS. Nevertheless, PS is the most accurate output time prediction approach if the related databases are continuingly updated to maintain enough validity, and often serves as a benchmark for evaluating the effectiveness of another method. PS also tends to be preferred because it allows for computational experiments and subsequent analyses without any actual execution [3]. Considering both effectiveness and efficiency, Chang et al. [4] and Chang and Hsieh [2] both forecasted the output/cycle time of a wafer lot with a BPN having a single hidden layer. Compared with MFLC approaches, the average prediction accuracy measured with the root mean squared error (RMSE) was considerably improved with these BPNs. For example, an improvement of about 40% in the RMSE was achieved in Chang et al. [4]. On the other hand, much less time and fewer data are required to generate an output time forecast with a BPN than with PS. More recently, Chang et al. [3] proposed a *k*-nearest-neighbours based case-based reasoning (CBR) approach which outperformed the BPN approach in forecasting accuracy. In one case, the advantage was up to 27%. Chang et al. [4] modified the first step (i.e. partitioning the range of each input variable into several fuzzy intervals) of the fuzzy modelling method proposed by Wang and Mendel [15], 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 wafer lot. Their EFR approach outperformed CBR and BPN in prediction accuracy. Chen [5] constructed a fuzzy BPN (FBPN) that incorporated expert opinions in forming inputs to the FBPN. Chen's FBPN was a hybrid approach (fuzzy modelling and BPN) and surpassed the crisp BPN especially in the efficiency respect.

According to these results, the concept of classifying inputs, which has been adopted in CBR and EFR, can indeed improve the effectiveness (prediction accuracy) of wafer lot output time prediction. This fact motivates us to design a similar hybrid approach – a hybrid artificial neural network (ANN) composed of a self-organization map (SOM) classifier and then a FBPN regression for the same purpose. On the other hand, all the aforementioned methods are based on the historical data of the fab. However, a lot of studies have shown that the performance of sequencing and scheduling in a fab relies heavily on the future release plan, which has been neglected in this field. In addition, the characteristic re-entrant production flows of a fab lead to the phenomenon that a lot that will be released in the future might appear in front of another lot that currently exists in the fab. For these reasons, to further improve the accuracy of wafer lot output time prediction, the future release plan of the fab has to be considered. Finally, a hybrid ANN (SOM+FBPN) incorporating the future release plan of the fab is proposed in this study. The system architecture is shown in Fig. 1.

PS is also applied in this study to generate test examples. Using simulated data, the effectiveness of the proposed methodology is shown and compared with those of three approaches, EFR, FBPN, and the hybrid ANN without considering the future release plan.

2 Methodology

2.1 Incorporating the Future Release Plan of the Fab

There are many possible ways to incorporate the future release plan in predicting the output time of a wafer lot currently existing in the fab. In this study, the three nearest future discounted workloads on the lot's processing route (according to the future release plan) are proposed for this purpose:

(a) *The 1st nearest future discounted workload* (*FDW_1*): the sum of the (processing time/release time)'s of the operations of the lots that will be released within time [now, now + T_1].

(b) *The 2nd nearest future discounted workload* (*FDW_2*): the sum of the (processing time/release time)'s of the operations of the lots that will be released within time [now + T_1, now + T_1 + T_2].

(c) *The 3rd nearest future discounted workload* (*FDW_3*): the sum of the (processing time/release time)'s of the operations of the lots that will be released within time [now + T_1 + T_2, now + T_1 + T_2 + T_3].

Note that only the operations performed on the machines on the lot's processing route are considered in calculating these future workloads, which then become three additional inputs to the FBPN.

2.2 Example Classification with SOM

In this study, a hybrid ANN (SOM+FBPN) incorporating the future release plan of the fab is proposed to predict the output time of a wafer lot. The first part of the hybrid ANN is a SOM classifier used to cluster examples. The reasons for adopting a SOM classifier instead of the others include:

(1) SOM has been proven useful in many applications including clustering, classification, monitoring, data visualization, etc., and is one of the most popular neural networks used in unsupervised learning.

(2) SOM can serve as a clustering tool for high dimensional data (e.g. production data in a wafer fab).

(3) There is potential for combination between SOM and another artificial neural network.

Every lot fed into the FBPN is called an example. Examples are pre-classified into different categories before they are fed into the FBPN with SOM. Let X={x_1, x_2, . . . , x_n} denote the set of feature vectors corresponding to the examples. Each item x_i is a nine-dimensional feature vector whose elements are the FDW_1, FDW_2, FDW_3, U_n, Q_n, BQ_n, FQ_n, WIP_n, and $D_n^{(i)}$ of the corresponding example. These feature vectors are fed into an SOM network. After the training is accomplished, input vectors that are topologically close are mapped to the same category, which means the input space is divided into *k* categories, and each example is associated with a certain category. Then, the classification result is post-processed, including eliminating isolated examples, merging small blocks, etc. Finally, the classification is finished.

After classification, examples of different categories are then learned with the same FBPN but with different parameter values.

2.3 FBPN for Output Time Prediction within Each Category

The configuration of the FBPN is established as follows:

(1) Inputs: nine parameters associated with the *n*-th example/lot including the average fab utilization (U_n), the total queue length on the lot's processing route (Q_n) or before bottlenecks (BQ_n) or in the whole fab (FQ_n), the fab WIP (WIP_n), the latenesses ($D_n^{(i)}$) of the *i*-th recently completed lots, and the three nearest future discounted workloads on the lot's processing route (FDW_1, FDW_2, and FDW_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 (output) time of a wafer 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, and the fuzzy arithmetic for triangular fuzzy numbers 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 network.

(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. [3]).

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

(5) Network learning rule: Delta rule.

(6) Transformation function: Sigmoid function,

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

(7) Learning rate (η): 0.01~1.0.

(8) Batch learning.

The procedure for determining the parameter values is now described. After pre-classification, a portion of the adopted examples in each category is fed as "training examples" into the FBPN to determine the parameter values 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 as:

$$\tilde{h}_j = (h_{j1},\ h_{j2},\ h_{j3}) = \frac{1}{1+e^{-\tilde{n}_j^h}} \tag{2}$$

where

$$\tilde{n}_j^h = (n_{j1}^h,\ n_{j2}^h,\ n_{j3}^h) = \tilde{I}_j^h(-)\tilde{\theta}_j^h \tag{3}$$

$$\tilde{I}_j^h = (I_{j1}^h,\ I_{j2}^h,\ I_{j3}^h) = \sum_{all\ \ i} \tilde{w}_{ij}^h(\times)\tilde{x}_{(i)} \tag{4}$$

and (−) and (×) denote fuzzy subtraction and multiplication, respectively; $\tilde{h}_j$'s are also transferred to the output layer with the same procedure. Finally, the output of the FBPN is generated as:

$$\tilde{o} = (o_1,\ o_2,\ o_3) = \frac{1}{1+e^{-\tilde{n}^o}} \tag{5}$$

where

$$\tilde{n}^o = (n_1^o,\ n_2^o,\ n_3^o) = \tilde{I}^o(-)\tilde{\theta}^o \tag{6}$$

$$\tilde{I}^o = (I_1^o,\ I_2^o,\ I_3^o) = \sum_{all\ \ j} \tilde{w}_j^o(\times)\tilde{h}_j \tag{7}$$

To improve the practical applicability of the FBPN and to facilitate the comparisons with conventional techniques, the fuzzy-valued output $\tilde{o}$ is defuzzified according to the centroid-of-area (COA) formula:

$$o = \text{COA}(\tilde{o}) = \frac{o_1 + 2o_2 + o_3}{4} \tag{8}$$

Then the defuzzified output *o* is applied to predict the actual cycle time *a*, for which the RMSE is calculated:

$$RMSE = \sqrt{\frac{\sum_{\text{all examples}} (o-a)^2}{\text{number of examples}}} \tag{9}$$

Subsequently in the backward phase, the deviation between *o* and *a* is propagated backward, and the error terms of neurons in the output and hidden layers can be calculated, respectively, as

$$\delta^o = o(1-o)(a-o) \tag{10}$$

$$\tilde{\delta}_j^h = (\delta_{j1}^h,\ \delta_{j2}^h,\ \delta_{j3}^h) = \tilde{h}_j(\times)(1-\tilde{h}_j)(\times)\tilde{w}_j^o\delta^o \tag{11}$$

Based on them, adjustments that should be made to the connection weights and thresholds can be obtained as

$$\Delta\tilde{w}_j^o = (\Delta w_{j1}^o,\ \Delta w_{j2}^o,\ \Delta w_{j3}^o) = \eta\delta^o\tilde{h}_j \tag{12}$$

$$\Delta\tilde{w}_{ij}^h = (\Delta w_{ij1}^h,\ \Delta w_{ij2}^h,\ \Delta w_{ij3}^h) = \eta\tilde{\delta}_j^h(\times)\tilde{x}_i \tag{13}$$

$$\Delta\theta^o = -\eta\delta^o \tag{14}$$

$$\Delta\tilde{\theta}_j^h = (\Delta\theta_{j1}^h,\ \Delta\theta_{j2}^h,\ \Delta\theta_{j3}^h) = -\eta\tilde{\delta}_j^h \tag{15}$$

Theoretically, network-learning stops when the RMSE falls below a pre-specified level, or the improvement in the RMSE becomes negligible with more epochs, or a large number of epochs have already been run. Then test examples are fed into the FBPN to evaluate the accuracy of the network that is also measured with the RMSE. However, the accumulation of fuzziness during the training process continuously increases the lower bound, the upper bound, and the spread of the fuzzy-valued output $\tilde{o}$ (and those of many other fuzzy parameters), and might prevent the RMSE (calculated with the defuzzified output *o*) from converging to its minimal value. Conversely, the centers of some fuzzy parameters are becoming smaller and smaller because of network learning. It is possible that a fuzzy parameter becomes invalid in the sense that the lower bound higher than the center. To deal with this problem, the lower and upper bounds of all fuzzy numbers in the FBPN will no longer be modified if Chen's index [5] converges to a minimal value.

Finally, the FBPN can be applied to predicting the cycle time of a new lot. When a new lot is released into the fab, the nine parameters associated with the new lot are recorded and compared with those of each category center. Then the FBPN with the parameters of the nearest category center is applied to forecasting the cycle time of the new lot. In this study, the SOM was implemented on the software "NeuroSolutions 4.0", while a VB.NET program has been constructed to implement the FBPN.

3 PS for Generating Test Data

In practical situations, the history data of each lot is only partially available in the factory. Further, some information of the previous lots such as Q_n, BQ_n, and FQ_n is not easy to collect on the shop floor. Therefore, a simulation model is often built to simulate the manufacturing process of a real wafer fabrication factory [1-5, 8, 11]. Then, such information can be derived from the shop floor status collected from the simulation model [3]. To generate test data, a simulation program coded using Microsoft Visual Basic .NET is constructed to simulate a wafer fabrication environment with the following assumptions:

(1) The distributions of the interarrival times of orders are exponential.

(2) The distributions of the interarrival times of machine downs are exponential.

(3) The distribution of the time required to repair a machine is deterministic.

(4) 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. However, the product mix in the simulated fab does fluctuate and is only approximately fixed in the long term.

(5) The percentages of lots with different priorities released into the fab are controlled.

(6) The priority of a lot cannot be changed during fabrication.

(7) Lots are sequenced on each machine first by their priorities, then by the first-in-first-out (FIFO) policy. Such a sequencing policy is a common practice in many foundry fabs.

(8) A lot has equal chances to be processed on each alternative machine/head available at a step.

(9) A lot cannot proceed to the next step until the fabrication on its every wafer has been finished.

(10) No preemption is allowed.

The basic configuration of the simulated wafer fab is the same as a real-world wafer fabrication factory which is located in the Science Park of Hsin-Chu, Taiwan, R.O.C. A trace report was generated every simulation run for verifying the simulation model. The simulated average cycle times have also been compared with the actual values to validate the simulation model, and the deviations were considered small. Assumptions (1)~(3), and (7)~(9) are commonly adopted in related researches (e.g. [2-5]), while assumptions (4)~(6) 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%). POs with normally distributed sizes (mean = 300 wafers; standard deviation = 50 wafers) arrive according to a Poisson process, and then the corresponding MOs are released for these POs a fixed time after. Based on these assumptions, the mean inter-release time of MOs into the fab can be obtained as (30.5 * 24) / (20000 / 300) = 11 hours. An MO is split into lots of a standard size of 24 wafers per lot. Lots of the same MO are released one by one every 11 / (300/24) = 0.85 hours. Three types of priorities (normal lot, hot lot, and super hot lot) 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 example. It also shows the difficulty for the production planning and scheduling people to provide an accurate due-date for the product with such a complicated routing. Totally 102 machines (including alternative machines) are provided to process single-wafer or batch operations in the fab. Thirty replicates of the simulation are successively run. The time required for each simulation replicate is about 12 minute on a PC with 512MB RAM and Athlon™ 64 Processor 3000+ CPU. A horizon of twenty-four months is simulated. The maximal cycle time is less than three months. Therefore, four 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 replicate, data of 30 lots are collected and classified by their product types and priorities. Totally, 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.

After drawing the time series plot of 100 simulated cycle time data, we observed that the pattern of the cycle time was not stable and very non-stationary. The traditional approach by human decision is very inaccurate and very prone to failure when the shop status is totally different even for the same product.

4 Results and Discussions

To evaluate the effectiveness and efficiency of the proposed methodology (indicated with HANNw) and to make some comparisons with three approaches – FBPN, EFR, and the hybrid ANN without considering the future release plan (indicated with HANNw/o), all the four methods were applied to five test cases containing the data of full-size (24 wafers per lot) lots with different product types and priorities. The minimal RMSEs achieved by applying the four approaches to different cases were recorded and compared in Table 1. The convergence condition was established as either the improvement in the RMSE becomes less than 0.001 with one more epoch, or 1000 epochs have already been run. According to experimental results, the following discussions are made:

(1) From the effectiveness viewpoint, the prediction accuracy (measured with the RMSE) of the proposed HANNw was significantly better than those of the other approaches in all cases by achieving a 20%~49% (and an average of 35%) reduction in the RMSE over the comparison basis – the FBPN. The average advantage over EFR is 8%.

(2) The effect of incorporating the future release plan of the fab is revealed by the fact that the prediction accuracy of HANNw was considerably better than that of HANNw/o in all cases with an average advantage of 4%.

(3) In the case that the lot priority was the highest (super hot lot), the proposed HANNw has the greatest advantage over FBPN in forecasting accuracy. In fact, the cycle time variation of super hot lots is the smallest, which makes their cycle times easy to predict. Clustering such lots seems to provide the most significant effect on the performance of cycle time prediction.

(4) As the lot priority increases, the superiority of HANNw over FBPN becomes more evident.

5 Conclusion and Directions for Future Research

Traditional wafer lot output time prediction methods are based on the historical data of the fab. The influence of the future release plan has been neglected. In addition, a lot that will be released in the future might appear in front of another lot that currently exists in the fab. For these reasons, to further improve the accuracy of wafer lot output time prediction, the future release plan of the fab has to be considered, and a hybrid ANN (SOM+FBPN) incorporating the future release plan of the fab is proposed in this study. For evaluating the effectiveness of the proposed approach and to make some comparisons with three approaches – FBPN, EFR, and the hybrid ANN without considering the future release plan, production simulation is applied in this study to generate test data. Then all the four methods are applied to five cases elicited from the test data. According to experimental results, the prediction accuracy (measured with the RMSE) of the proposed approach was significantly better than those of the other approaches in all cases by achieving a 20%~49% (and an average of 35%) reduction in the RMSE over the comparison basis – the FBPN. The average advantage over EFR is 8%. The effect of incorporating the future release plan of the fab is also evident.

However, to further evaluate the effectiveness and efficiency of the proposed methodology, it has to be applied to fab models of different scales, especially a full-scale actual wafer fab. In addition, the proposed methodology can also be applied to cases with changing product mixes or loosely controlled priority combinations, under which the cycle time variation is often very large.

At last, other traditional approaches can also be improved by incorporating the future release plan of the fab. There are also many other possible ways to incorporate the future release plan of the fab. These constitute two directions for future research.

References:

[1] S. Barman, The impact of priority rule combinations on lateness and tardiness, IIE Transactions, Vol.30, 1998, pp. 495-504.

[2] P.-C. Chang, J.-C. Hsieh, A neural networks approach for due-date assignment in a wafer fabrication factory, International Journal of Industrial Engineering, Vol.10, No.1, 2003, pp. 55-61.

[3] P.-C. Chang, J.-C. Hsieh, T. W. Liao, A case-based reasoning approach for due date assignment in a wafer fabrication factory, in: Proceedings of the International Conference on Case-Based Reasoning (ICCBR 2001), Vancouver, British Columbia, Canada, 2001.

[4] P.-C. Chang, J.-C. Hsieh, T. W. Liao, Evolving fuzzy rules for due-date assignment problem in semiconductor manufacturing factory, Journal of Intelligent Manufacturing, Vol.16, 2005, pp. 549-557.

[5] T. Chen, A fuzzy back propagation network for output time prediction in a wafer fab, Journal of Applied Soft Computing, Vol.2/3F, 2003, pp. 211-222.

[6] S.-H. Chung, M.-H. Yang, C.-M. Cheng, The design of due date assignment model and the determination of flow time control parameters for the wafer fabrication factories, IEEE Transactions On Components, Packaging, and Manufacturing Technology – Part C, Vol.20, No.4, 1997, pp. 278-287.

[7] W. R. Foster, F. Gollopy, L. H. Ungar, Neural network forecasting of short, noisy time series, Computers in Chemical Engineering, Vol.16, No.4, 1992, pp. 293-297.

[8] D. E. Goldberg, Genetic Algorithms in Search, Optimization, and Machine Learning, Addison-Wesley, Reading, MA, 1989.

[9] Y.-F. Hung, C.-B. Chang, Dispatching rules using flow time predictions for semiconductor wafer fabrications, in: Proceedings of the 5th Annual International Conference on Industrial Engineering Theory, Applications and Practice, Taiwan, 2001.

[10] Y. Jiang, Z. H. Zhou, SOM ensemble-based image segmentation, Neural Processing Letters, Vol.20, 2004, pp. 171-178.

[11] C.-Y. Lin, Shop floor scheduling of semiconductor wafer fabrication using real-time feedback control and prediction, Ph.D. Dissertation, Engineering-Industrial Engineering and Operations Research, University of California at Berkeley, 1996.

[12] S. Piramuthu, Theory and methodology – financial credit-risk evaluation with neural and neuralfuzzy systems, European Journal of Operational Research, Vol.112, 1991, pp. 310-321.

[13] G. L. Ragatz, V. A. Mabert, A simulation analysis of due date assignment, Journal of Operations Management, Vol.5, 1984, pp. 27-39.

[14] M. M. Vig, K. J. Dooley, Dynamic rules for due-date assignment, International Journal of Production Research, Vol.29, No.7, 1991, pp. 1361-1377.

[15] L.-X. Wang, J. M. Mendel, Generating fuzzy rules by learning from examples, IEEE Transactions on Systems, Man, and Cybernetics, Vol.22, No.6, 1992, pp. 1414-1427.

[16] J. K. Weeks, A simulation study of predictable due-dates, Management Science, Vol.25, 1979, pp. 363–373.

Table 1. Comparisons of the RMSEs of various approaches

RMSE	FBPN	EFR	HANNw/o	HANNw
A(normal lots)	177.1	164.29 (-7%)	151.34 (-15%)	141.47 (-20%)
A(hot lots)	102.27	66.21 (-35%)	63.66 (-38%)	59.51 (-42%)
A(super hot lots)	12.23	9.07 (-26%)	9.72 (-21%)	9.07 (-26%)
B(normal lots)	286.93	208.28 (-27%)	188.55 (-34%)	178.42 (-38%)
B(hot lots)	75.98	44.57 (-41%)	41.43 (-45%)	38.59 (-49%)

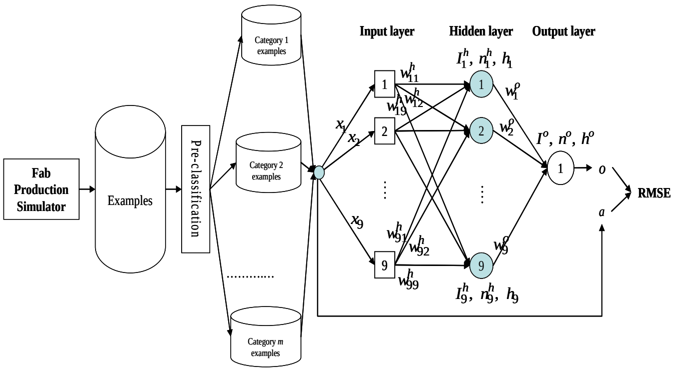

Fig. 1. System architecture