


Article

A Selectively Fuzzified Back Propagation Network Approach for Precisely Estimating the Cycle Time Range in Wafer Fabrication

Yu-Cheng Wang ¹, Horng-Ren Tsai ² and Toly Chen ^{3,*}¹ Department of Aeronautical Engineering, Chaoyang University of Technology, Taichung 41349, Taiwan; tony.cobra@msa.hinet.net² Department of Information Technology, Lingtung University, Taichung 408213, Taiwan; hrt@teamail.ltu.edu.tw³ Department of Industrial Engineering and Management, National Yang Ming Chiao Tung University, 1001, University Road, Hsinchu 30010, Taiwan

* Correspondence: tolychen@ms37.hinet.net

Abstract: Forecasting the cycle time of each job is a critical task for a factory. However, recent studies have shown that it is a challenging task, even with state-of-the-art deep learning techniques. To address this challenge, a selectively fuzzified back propagation network (SFBPN) approach is proposed to estimate the range of a cycle time, the results of which provide valuable information for many managerial purposes. The SFBPN approach is distinct from existing methods, because the thresholds on both the hidden and output layers of a back propagation network are fuzzified to tighten the range of a cycle time, while most of the existing methods only fuzzify the threshold on the output node. In addition, a random search and local optimization algorithm is also proposed to derive the optimal values of the fuzzy thresholds. The proposed methodology is applied to a real case from the literature. The experimental results show that the proposed methodology improved the forecasting precision by up to 65%.

Keywords: cycle time; forecasting; selectively fuzzified back propagation network; fuzzy collaborative forecasting



Citation: Wang, Y.-C.; Tsai, H.-R.; Chen, T. A Selectively Fuzzified Back Propagation Network Approach for Precisely Estimating the Cycle Time Range in Wafer Fabrication.

Mathematics **2021**, *9*, 1430. <https://doi.org/10.3390/math9121430>

Academic Editor: Alfonso Mateos Caballero

Received: 31 May 2021
Accepted: 16 June 2021
Published: 19 June 2021

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2021 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

This study aims to estimate the cycle time range of a job in a factory. The cycle time (or manufacturing lead time) of a job is the time it takes for the job to pass through the factory [1]. Various methods have been proposed to predict the cycle time of a job in a factory [2–4]. However, even if some advanced computing techniques, such as big data or deep learning, are applied, the prediction accuracy is still not good enough [3–6]. To overcome this problem, estimating the range of the cycle time instead is a meaningful treatment.

Estimating the cycle time range is an important issue because it provides valuable information for various managerial activities. For example, in internal due date assignment [1,5], an internal due date must be later than the upper bound of the cycle time in order to ensure timely delivery [6,7]. However, it is also a challenging task because the cycle time of a job is subject to many uncertainties caused by unstable human intervention, unexpected machine breakdown, poor job sequencing and scheduling, and so on [8–10]. As a result, the cycle time range of a job may be very wide, or does not even contain the actual value, which represents poor estimation precision [4,11,12].

Based on the above reasons, the research problem of this study is on how to improve the precision for estimating the cycle time range of each job in a factory. Existing methods in this field are subject to the following problems:

- (1) Some existing methods establish the confidence interval of the cycle time [12–15]. However, even with advanced computing technologies such as deep learning and big data analysis, the accuracy of predicting the cycle time is still not satisfactory. As far

as the impact is reached, the established confidence interval of the cycle time has no reference value.

- (2) In some studies, the parameters of a cycle time forecasting method were fuzzified to generate a fuzzy forecast, which represents the range of the cycle time. Most of these methods only fuzzified a single parameter to simplify calculations [4,16]. However, fuzzifying more parameters can further shorten the range of the cycle time.

To solve the problems of the existing methods and to improve the precision of estimating the cycle time range of a job, a selectively fuzzified (SFBPN) approach is proposed in this study. The contribution of the proposed methodology is to establish a systematic procedure to efficiently fuzzify the multiple network parameters of a BPN, thereby further reducing the cycle time range of each job.

In the proposed methodology, a SFBPN is constructed to estimate the cycle time range of a job. In SFBPN, the thresholds on the hidden-layer and output-layer nodes are fuzzified to further tighten the cycle time range. To derive the optimal values of the fuzzy thresholds, a nonlinear programming (NLP) problem needs to be solved, which is not easy. To tackle this difficulty, a random search and local optimization algorithm is proposed.

The remainder of this paper is organized as follows. Section 2 is dedicated to the literature review. Section 3 introduces the proposed methodology, including the implementation procedure, the SFBPN architecture, and the random search and local optimization algorithm. To assess the effectiveness of the proposed methodology, it has been applied to a real case from the literature, which is described in Section 4. Section 5 presents concluding remarks and puts forth some topics for future investigation.

2. Literature Review

Chen and Wang [13] applied fuzzy c-means (FCM) to classify jobs in a factory, and then constructed a backpropagation network (BPN) to forecast the cycle times of the jobs for each category. The range of the cycle time was estimated by constructing the confidence interval of the cycle time. However, in theory, the confidence interval does not necessarily contain the actual value. To solve this problem, Chen and Lin [17] fuzzified the parameters of a BPN to generate a fuzzy cycle time forecast. The support of a fuzzy cycle time forecast represented the range of the cycle time. However, nonlinear programming (NLP) problems needed to be solved, which was not easy.

Hsieh et al. [18] applied response surface modelling (RSM) to evaluate the impact of emergency jobs on the cycle times of normal jobs. More emergency jobs extended the cycle times of normal jobs and widened the ranges of these cycle times. Therefore, the percentage of emergent jobs in the factory should be minimized to improve the precision of estimating the cycle ranges of normal jobs.

Wang and Zhang [14] modified the FCM–BPN method [13] by incorporating a conditional mutual information-based feature selection mechanism to choose the inputs for the FCM–BPN method. The improvement in the forecasting accuracy also helped to establish a narrower cycle time range. However, the range was based on the confidence interval, which might not contain the actual value.

In Chen [4], the threshold on the output node of a BPN was fuzzified to estimate the range of the cycle time. Then, fuzzy intersection (FI) was applied to aggregate the cycle time ranges estimated by multiple experts. In this way, the cycle time range could be further reduced. However, the cycle time range was affected by extreme cases, namely jobs with unexpected long- or short-cycle times [19]. To overcome this problem, Chen [20] slightly adjusted each cycle time forecast before fuzzifying the threshold on the output node. In this way, the impact of extreme cases could be mitigated. However, as only one network parameter was fuzzified, there was still room for improvement.

Wang et al. [21] constructed a two-dimensional long short-term memory (LSTM) network with multiple memory units to forecast the cycle time of a job. The LSTM network was a deep recurrent neural network [22,23]. Through deep learning and considering the correlation between the network parameters, the forecasting accuracy could be improved,

thereby helping to reduce the cycle time range. However, the problem of a confidence interval still existed.

Chen and Wu [16] replaced a cycle time forecast with its linear function before fuzzifying the threshold on the output node of a BPN to estimate the cycle time range, which further tightened the lower and upper bounds of the cycle time [17]. Fuzzifying only one network parameter limited the scope of improvement.

Wang et al. [24] modified the approach proposed by Wang and Zhang [14] by incorporating an adaptive logistic regression correlation analysis-based feature selection mechanism instead. These two studies suffered from the same problem of a confidence interval [25,26].

The novelty of the proposed methodology is highlighted by comparing it with some of the existing methods, as summarized in Table 1.

Table 1. The novelty of the proposed methodology.

Method	Type of Cycle Time Forecasts	Fuzzified Parameters	Optimization Method	Precision
Wang and Zhang [14]	Crisp	No	Optimized feature selection	Low
Chen and Wu [16]	Fuzzy	Threshold on the node of the output layer	Equations based on the linear function of the output	High
Wang et al. [21]	Crisp	No	Deep learning	Low
Wang et al. [24]	Crisp	No	Optimized feature selection	Low–moderate
The proposed methodology	Fuzzy	Thresholds on nodes of hidden and output layers	Random search and local optimization algorithm	Very high

3. Methodology

The implementation procedure of the proposed methodology comprised the following steps:

- (1) Preprocess the collected data: two major tasks in this step are feature selection and data normalization.
- (2) Construct a SFBPN to forecast the cycle time of a job.
- (3) Train the SFBPN using an existing algorithm to derive the cores of the network parameters.
- (4) Apply the random search and local optimization algorithm to derive the lower and upper bounds of the thresholds.
- (5) Estimate the cycle time ranges of all of the jobs.
- (6) Evaluate the forecasting precision.

A flow chart is provided in Figure 1 to illustrate the implementation procedure of the proposed methodology. Without a loss of generality, all of the fuzzy parameters and variables in the proposed methodology are given in or approximated with triangular fuzzy numbers (TFNs) [27,28].

3.1. Data Preprocessing

There are two major tasks at the stage of data preprocessing, namely, feature selection and data normalization.

First, relevant features are selected based on the way the cycle time of a job is forecasted. One way is to treat the cycle times of jobs as a time series and to apply a time series forecasting method to forecast the job cycle times [29], for which the relevant features are the cycle times of jobs that have been completed recently. The other way is to fit the relationship between the cycle time of a job and the attributes of the job. In this way, relevant features include the attributes of a job, the cycle times of jobs that have been completed recently, production conditions when a job is released into the wafer fab, etc. [30,31].

Features can also be functionalized, split, or combined using techniques such as principal component analysis and stepwise regression, before serving as inputs [24,32]. In the literature, various techniques have been applied to select relevant features, e.g., subjective elimination based on expert knowledge [24,33], backward elimination-based regression analysis [31], backward elimination-based genetic programming [34,35], conditional mutual information-based feature selection [14,21,36], adaptive logistic regression correlation analysis [24], mutual information network deconvolution feature selection [37], etc. In this study, input features were selected using backward elimination-based regression analysis.

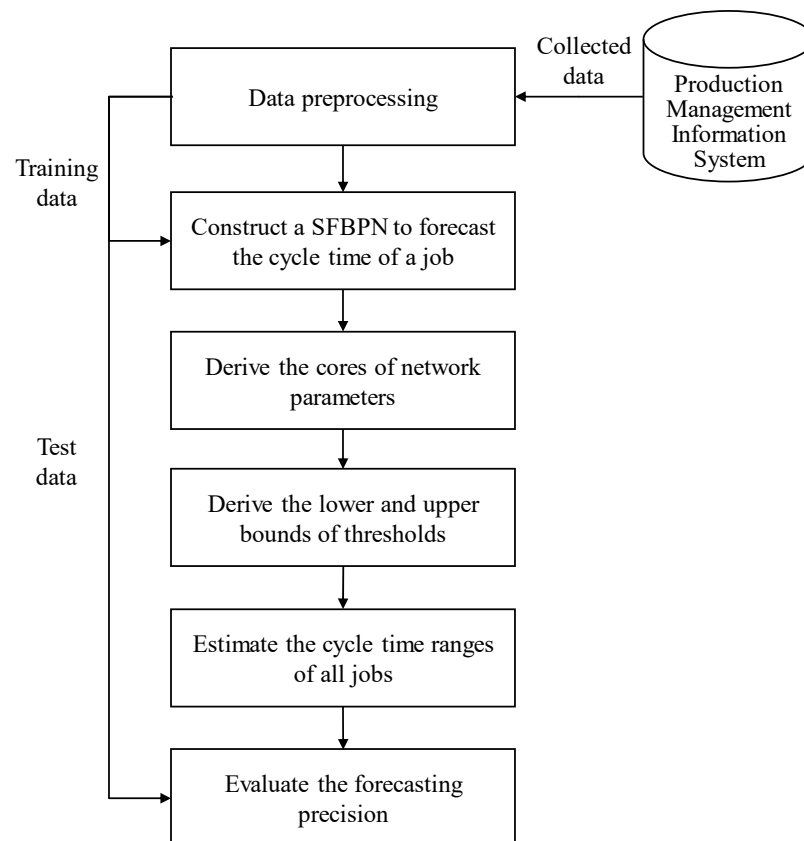


Figure 1. The implementation procedure of the proposed methodology.

Subsequently, the collected data were normalized into [0.1, 0.9] to ensure the extrapolation ability of the SFBPN using the partial normalization method [4]:

$$\begin{aligned}
 z_{jp} &= N(x_{jp}) \\
 &= \frac{x_{jp} - \min_r x_{rp}}{\max_r x_{rp} - \min_r x_{rp}} \cdot 0.8 + 0.1
 \end{aligned} \quad (1)$$

where $N()$ is the partial normalization function. j and r are both indexes of a job; $1 \leq j, r \leq n$. p is the index of an input; $1 \leq p \leq P$. x_{jp} (or x_{rp}) is the p -th attribute of job j (or r); z_{jp} is the normalized value of x_{jp} . To convert z_{jp} back to the original value:

$$\begin{aligned}
 x_{jp} &= U(z_{jp}) \\
 &= \frac{z_{jp} - 0.1}{0.8} \cdot (\max_r x_{rp} - \min_r x_{rp}) + \min_r x_{rp}
 \end{aligned} \quad (2)$$

where $U()$ is the partial un-normalization function.

3.2. Forecasting the Cycle Time of a Job Using a SFBPN

In the proposed methodology, a SFBPN is constructed to forecast the cycle time of a job in a wafer fab. The SFBPN is a special FBPN for which the parameters are selectively fuzzified. The architecture of the SFBPN is illustrated in Figure 2, which has three layers, namely: the input layer, the hidden layer, and the output layer.

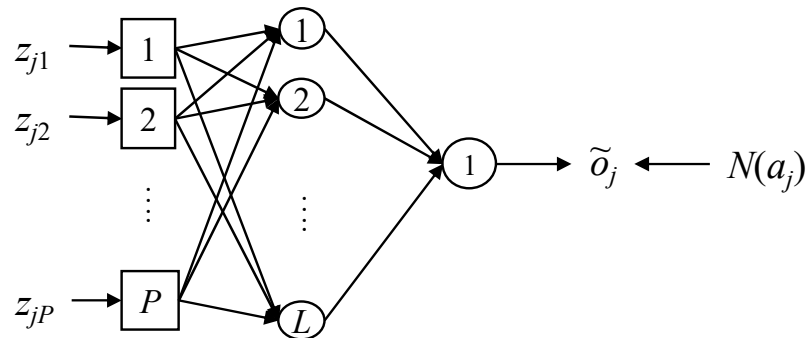


Figure 2. The architecture of the selectively fuzzified backpropagation network (SFBPN).

In the network training phase, inputs to the SFBPN are weighted and transmitted to each node of the hidden layer, on which they are aggregated and then outputted as follows

$$\begin{aligned}\tilde{h}_{jl} &= (h_{jl1}, h_{jl2}, h_{jl3}) \\ &= \frac{1}{1+e^{-\tilde{n}_{jl}^h}} \\ &\cong \left(\frac{1}{1+e^{-n_{jl1}^h}}, \frac{1}{1+e^{-n_{jl2}^h}}, \frac{1}{1+e^{-n_{jl3}^h}} \right)\end{aligned}\quad (3)$$

where

$$\begin{aligned}\tilde{n}_{jl}^h &= (n_{jl1}^h, n_{jl2}^h, n_{jl3}^h) \\ &= \tilde{I}_{jl}^h(-)\tilde{\theta}_l^h \\ &= (I_{jl1}^h - \theta_{l3}^h, I_{jl2}^h - \theta_{l2}^h, I_{jl3}^h - \theta_{l1}^h)\end{aligned}\quad (4)$$

$$\begin{aligned}\tilde{I}_{jl}^h &= (I_{jl1}^h, I_{jl2}^h, I_{jl3}^h) \\ &= \sum_{p=1}^P \tilde{w}_{pl}^h z_{jp} \\ &= \left(\sum_{p=1}^P w_{pl1}^h z_{jp}, \sum_{p=1}^P w_{pl2}^h z_{jp}, \sum_{p=1}^P w_{pl3}^h z_{jp} \right)\end{aligned}\quad (5)$$

where l is the index of a hidden-layer node; $l = 1 \sim L$. \tilde{h}_{jl} is the output from node l of the hidden layer. $\tilde{\theta}_l^h$ is the threshold on this node; \tilde{w}_{pl}^h is the weight of the connection between input node p and this node. $(-)$ denotes fuzzy subtraction [38]. After passing \tilde{h}_{jl} to the output layer, the network output \tilde{o}_j is generated as follows

$$\begin{aligned}\tilde{o}_j &= (o_{j1}, o_{j2}, o_{j3}) \\ &= \frac{1}{1+e^{-\tilde{n}_j^o}} \\ &\cong \left(\frac{1}{1+e^{-n_{j1}^o}}, \frac{1}{1+e^{-n_{j2}^o}}, \frac{1}{1+e^{-n_{j3}^o}} \right)\end{aligned}\quad (6)$$

where

$$\begin{aligned}\tilde{o}_j &= (o_{j1}, o_{j2}, o_{j3}) \\ &= \frac{1}{1+e^{-\tilde{n}_j^o}} \\ &\cong \left(\frac{1}{1+e^{-n_{j1}^o}}, \frac{1}{1+e^{-n_{j2}^o}}, \frac{1}{1+e^{-n_{j3}^o}} \right)\end{aligned}\quad (7)$$

$$\begin{aligned}
\tilde{I}_j^o &= (I_{j1}^o, I_{j2}^o, I_{j3}^o) \\
&= \sum_{l=1}^M \tilde{w}_l^o (\times) \tilde{h}_{jl} \\
&\cong \left(\sum_{l=1}^M \min(w_{l1}^o h_{jl1}, w_{l1}^o h_{jl3}, w_{l3}^o h_{jl1}, w_{l3}^o h_{jl3}), \sum_{l=1}^M w_{l2}^o h_{jl2}, \right. \\
&\quad \left. \sum_{l=1}^M \max(w_{l1}^o h_{jl1}, w_{l1}^o h_{jl3}, w_{l3}^o h_{jl1}, w_{l3}^o h_{jl3}) \right)
\end{aligned} \tag{8}$$

$\tilde{\theta}^o$ is the threshold on the output node; \tilde{w}_l^o is the weight of the connection between node l of the hidden layer and the output node. As $\tilde{h}_{jl} \geq 0$, Equation (8) can be shortened, as follows

$$\tilde{I}_j^o = \left(\sum_{\tilde{w}_l^o \geq 0} w_{l1}^o h_{jl1} + \sum_{\tilde{w}_l^o < 0} w_{l1}^o h_{jl3}, \sum_{l=1}^M w_{l2}^o h_{jl2}, \sum_{\tilde{w}_l^o \geq 0} w_{l3}^o h_{jl3} + \sum_{\tilde{w}_l^o < 0} w_{l3}^o h_{jl1} \right) \tag{9}$$

Some theoretical properties of the SFBPN are discussed in the following.

Theorem 1.

$$(I_{j1}^o, I_{j2}^o, I_{j3}^o) = (\theta_3^o - \ln(\frac{1}{o_{j1}} - 1), \theta_2^o - \ln(\frac{1}{o_{j2}} - 1), \theta_1^o - \ln(\frac{1}{o_{j3}} - 1)) \tag{10}$$

Proof. The required proof is trivial. \square

Property 1. o_{j3} decreases when θ_1^o increases; o_{j1} increases when θ_3^o decreases.

3.3. Determining the Values of Network Parameters

The training of the SFBPN is decomposed into three tasks, i.e., determining the core, lower bound, and upper bound of each network parameter [39].

At first, the SFBPN is treated as a crisp one and trained using an existing algorithm, such as the gradient descent (GD) algorithm, the Levenberg–Marquardt (LM) algorithm, the Broyden–Fletcher–Goldfarb–Shanno (BFGS) quasi-Newton algorithm, the GD algorithm with momentum and adaptive learning rate (GDX), or the resilient backpropagation (RP) algorithm [40], to determine the cores of the network parameters (such as w_{pl2}^h , θ_{l2}^h , w_{l2}^o , and θ_2^o), thereby optimizing the forecasting accuracy measured in terms of the root mean squared error (RMSE):

$$RMSE = \sqrt{\frac{\sum_{i=1}^n (o_{j2} - N(a_j))^2}{n}} \tag{11}$$

where a_j is the cycle time (i.e., actual value) of job j . Assuming the obtained optimal solution is indicated with $\{w_{pl2}^{h*}, \theta_{l2}^{h*}, w_{l2}^{o*}, \theta_2^{o*}\}$.

Subsequently, the following NLP problem is solved to derive the lower and upper bounds of the network parameters, thereby optimizing the forecasting precision measured in terms of the average range (AR) of fuzzy cycle time forecasts:

(NLP Problem)

$$\text{Min } AR = \frac{1}{n} \sum_{j=1}^n (o_{j3} - o_{j1}) \tag{12}$$

subject to

$$o_{j1} \leq N(a_j); j = 1 \sim n \tag{13}$$

$$o_{j3} \geq N(a_j); j = 1 \sim n \tag{14}$$

$$o_{j1} = \frac{1}{1 + e^{-n_{j1}^o}}; j = 1 \sim n \quad (15)$$

$$o_{j3} = \frac{1}{1 + e^{-n_{j3}^o}}; j = 1 \sim n \quad (16)$$

$$n_{j1}^o = I_{j1}^o - \theta_3^o; j = 1 \sim n \quad (17)$$

$$n_{j3}^o = I_{j3}^o - \theta_1^o; j = 1 \sim n \quad (18)$$

$$I_{j1}^o = \sum_{w_{l2}^{o*} \geq 0} w_{l1}^o h_{jl1} + \sum_{w_{l2}^{o*} < 0} w_{l1}^o h_{jl3}; j = 1 \sim n \quad (19)$$

$$I_{j3}^o = \sum_{w_{l2}^{o*} \geq 0} w_{l3}^o h_{jl3} + \sum_{w_{l2}^{o*} < 0} w_{l3}^o h_{jl1}; j = 1 \sim n \quad (20)$$

$$h_{jl1} = \frac{1}{1 + e^{-n_{jl1}^h}}; j = 1 \sim n \quad (21)$$

$$h_{jl3} = \frac{1}{1 + e^{-n_{jl3}^h}}; j = 1 \sim n \quad (22)$$

$$n_{jl1}^h = I_{jl1}^h - \theta_{l3}^h; j = 1 \sim n \quad (23)$$

$$n_{jl3}^h = I_{jl3}^h - \theta_{l1}^h; j = 1 \sim n \quad (24)$$

$$I_{jl1}^h = \sum_{p=1}^P w_{pl1}^h z_{jp}; j = 1 \sim n \quad (25)$$

$$I_{jl3}^h = \sum_{p=1}^P w_{pl3}^h z_{jp}; j = 1 \sim n \quad (26)$$

$$o_{j1} \leq o_{j2}^* \leq o_{j3}; j = 1 \sim n \quad (27)$$

$$\theta_1^o \leq \theta_2^{o*} \leq \theta_3^o \quad (28)$$

$$w_{l1}^o \leq w_{l2}^{o*} \leq w_{l3}^o; l = 1 \sim L \quad (29)$$

$$\theta_{l1}^h \leq \theta_{l2}^{h*} \leq \theta_{l3}^h; l = 1 \sim L \quad (30)$$

$$w_{pl1}^h \leq w_{pl2}^{h*} \leq w_{pl3}^h; p = 1 \sim P; l = 1 \sim L \quad (31)$$

Constraints (30) to (33) define the sequence of the three corners of the corresponding TFN.

The NLP problem is not easy to solve. To tackle this difficulty, Chen and Lin [18] fuzzified only the threshold on the output node, and set the other network parameters to crisp values to simplify the problem, as follows:

$$w_{l1}^o = w_{l2}^{o*} = w_{l3}^o; l = 1 \sim L \quad (32)$$

$$\theta_{l1}^h = \theta_{l2}^{h*} = \theta_{l3}^h; l = 1 \sim L \quad (33)$$

$$w_{pl1}^h = w_{pl2}^{h*} = w_{pl3}^h; p = 1 \sim P; l = 1 \sim L \quad (34)$$

As a result, $I_{jl1}^h = I_{jl2}^{h*} = I_{jl3}^h$, $h_{jl1} = h_{jl2}^* = h_{jl3}$, and $I_{j1}^o = I_{j2}^{o*} = I_{j3}^o$. The threshold on the output node can be optimized as follows.

Theorem 2. Reference [17].

$$\theta_3^{o*} = \theta_2^{o*} + \min_j (\ln(\frac{1}{N(a_j)} - 1) - \ln(\frac{1}{o_j} - 1)) \quad (35)$$

$$\theta_1^{o*} = \theta_2^{o*} - \max_j (\ln(\frac{1}{N(a_j)} - 1) - \ln(\frac{1}{o_j} - 1)) \quad (36)$$

Proof. Substituting Equations (17) and (18) into Equations (15) and (16) gives the following

$$o_{j1} = \frac{1}{1 + e^{-(I_{j1}^o - \theta_3^o)}} \quad (37)$$

$$o_{j3} = \frac{1}{1 + e^{-(I_{j3}^o - \theta_1^o)}} \quad (38)$$

Substituting Equations (37) and (38) into Constraints (13) and (14), gives the following:

$$\frac{1}{1 + e^{-(I_{j1}^o - \theta_3^o)}} \leq N(a_j) \quad (39)$$

$$\frac{1}{1 + e^{-(I_{j3}^o - \theta_1^o)}} \geq N(a_j) \quad (40)$$

Therefore,

$$\theta_3^o \geq I_{j1}^o + \ln(\frac{1}{N(a_j)} - 1) \quad (41)$$

$$\theta_1^o \leq I_{j3}^o + \ln(\frac{1}{N(a_j)} - 1) \quad (42)$$

As $I_{j1}^o = I_{j2}^{o*} = I_{j3}^o$,

$$\theta_3^o \geq I_{j2}^{o*} + \ln(\frac{1}{N(a_j)} - 1) \quad (43)$$

$$\theta_1^o \leq I_{j2}^{o*} + \ln(\frac{1}{N(a_j)} - 1) \quad (44)$$

Applying Theorem 1 to Constraints (43) and (44) gives the following:

$$\theta_3^o \geq \theta_2^{o*} - \ln(\frac{1}{o_{j2}^*} - 1) + \ln(\frac{1}{N(a_j)} - 1) \quad (45)$$

$$\theta_1^o \leq \theta_2^{o*} - \ln(\frac{1}{o_{j2}^*} - 1) + \ln(\frac{1}{N(a_j)} - 1) \quad (46)$$

Constraints (45) and (46) hold for all jobs. Therefore,

$$\theta_3^o \geq \max_j (\theta_2^{o*} - \ln(\frac{1}{o_{j2}^*} - 1) + \ln(\frac{1}{N(a_j)} - 1)) \quad (47)$$

$$\theta_1^o \leq \min_j (\theta_2^{o*} - \ln(\frac{1}{o_{j2}^*} - 1) + \ln(\frac{1}{N(a_j)} - 1)) \quad (48)$$

To minimize $o_{j3} - o_{j1}$, o_{j3} and o_{j1} are to be minimized and maximized, respectively. For this purpose, according to Property 1, θ_1^o and θ_3^o should be maximized and minimized, respectively. As a result,

$$\theta_3^{o*} = \max_j (\theta_2^{o*} - \ln(\frac{1}{o_{j2}^*} - 1) + \ln(\frac{1}{N(a_j)} - 1)) \quad (49)$$

$$\theta_1^{o*} = \min_j (\theta_2^{o*} - \ln(\frac{1}{o_{j2}^*} - 1) + \ln(\frac{1}{N(a_j)} - 1)) \quad (50)$$

Theorem 2 is proved. \square

3.4. Fuzzifying Thresholds on Hidden-Layer Nodes

In the proposed methodology, the thresholds on both the hidden-layer and output-layer nodes are fuzzified to further enhance the forecasting precision measured in terms of the average range of fuzzy cycle time forecasts. The optimal values of these thresholds can be derived as follows.

Theorem 3.

$$\theta_3^{o*} = \min_j \left(\sum_{w_{l2}^{o*} \geq 0} \frac{w_{l2}^{o*}}{1 + e^{-(I_{jl2}^{h*} - \theta_{l3}^{h*})}} + \sum_{w_{l2}^{o*} < 0} \frac{w_{l2}^{o*}}{1 + e^{-(I_{jl2}^{h*} - \theta_{l1}^{h*})}} + \ln\left(\frac{1}{N(a_j)} - 1\right) \right) \quad (51)$$

$$\theta_1^{o*} = \max_j \left(\sum_{w_{l2}^{o*} \geq 0} \frac{w_{l2}^{o*}}{1 + e^{-(I_{jl2}^{h*} - \theta_{l1}^{h*})}} + \sum_{w_{l2}^{o*} < 0} \frac{w_{l2}^{o*}}{1 + e^{-(I_{jl2}^{h*} - \theta_{l3}^{h*})}} + \ln\left(\frac{1}{N(a_j)} - 1\right) \right) \quad (52)$$

Proof. First, substituting Equations (23) and (24) into Equations (21) and (22) gives the following

$$h_{jl1} = \frac{1}{1 + e^{-(I_{jl1}^h - \theta_{l3}^h)}}; j = 1 \sim n \quad (53)$$

$$h_{jl3} = \frac{1}{1 + e^{-(I_{jl3}^h - \theta_{l1}^h)}}; j = 1 \sim n \quad (54)$$

Then, substituting Equations (53) and (54) into Constraints (19) and (20) gives the following

$$I_{j1}^o = \sum_{w_{l2}^{o*} \geq 0} \frac{w_{l1}^o}{1 + e^{-(I_{jl1}^h - \theta_{l3}^h)}} + \sum_{w_{l2}^{o*} < 0} \frac{w_{l1}^o}{1 + e^{-(I_{jl3}^h - \theta_{l1}^h)}}; j = 1 \sim n \quad (55)$$

$$I_{j3}^o = \sum_{w_{l2}^{o*} \geq 0} \frac{w_{l3}^o}{1 + e^{-(I_{jl3}^h - \theta_{l1}^h)}} + \sum_{w_{l2}^{o*} < 0} \frac{w_{l3}^o}{1 + e^{-(I_{jl1}^h - \theta_{l3}^h)}}; j = 1 \sim n \quad (56)$$

$$\text{As } I_{jl1}^h = I_{jl2}^{h*} = I_{jl3}^h,$$

$$I_{j1}^o = \sum_{w_{l2}^{o*} \geq 0} \frac{w_{l2}^{o*}}{1 + e^{-(I_{jl2}^{h*} - \theta_{l3}^{h*})}} + \sum_{w_{l2}^{o*} < 0} \frac{w_{l2}^{o*}}{1 + e^{-(I_{jl2}^{h*} - \theta_{l1}^{h*})}}; j = 1 \sim n \quad (57)$$

$$I_{j3}^o = \sum_{w_{l2}^{o*} \geq 0} \frac{w_{l2}^{o*}}{1 + e^{-(I_{jl2}^{h*} - \theta_{l1}^{h*})}} + \sum_{w_{l2}^{o*} < 0} \frac{w_{l2}^{o*}}{1 + e^{-(I_{jl2}^{h*} - \theta_{l3}^{h*})}}; j = 1 \sim n \quad (58)$$

Substituting Equations (57) and (58) into Constraints (43) and (44) results in the following

$$\theta_3^o \geq \sum_{w_{l2}^{o*} \geq 0} \frac{w_{l2}^{o*}}{1 + e^{-(I_{jl2}^{h*} - \theta_{l3}^{h*})}} + \sum_{w_{l2}^{o*} < 0} \frac{w_{l2}^{o*}}{1 + e^{-(I_{jl2}^{h*} - \theta_{l1}^{h*})}} + \ln\left(\frac{1}{N(a_j)} - 1\right) \quad (59)$$

$$\theta_1^o \leq \sum_{w_{l2}^{o*} \geq 0} \frac{w_{l2}^{o*}}{1 + e^{-(I_{jl2}^{h*} - \theta_{l1}^{h*})}} + \sum_{w_{l2}^{o*} < 0} \frac{w_{l2}^{o*}}{1 + e^{-(I_{jl2}^{h*} - \theta_{l3}^{h*})}} + \ln\left(\frac{1}{N(a_j)} - 1\right) \quad (60)$$

According to Property 1,

$$\theta_3^{o*} = \min_j \left(\sum_{w_{l2}^{o*} \geq 0} \frac{w_{l2}^{o*}}{1 + e^{-(I_{jl2}^{h*} - \theta_{l3}^{h*})}} + \sum_{w_{l2}^{o*} < 0} \frac{w_{l2}^{o*}}{1 + e^{-(I_{jl2}^{h*} - \theta_{l1}^{h*})}} + \ln\left(\frac{1}{N(a_j)} - 1\right) \right) \quad (61)$$

$$\theta_1^{o*} = \max_j \left(\sum_{w_{l2}^{o*} \geq 0} \frac{w_{l2}^{o*}}{1 + e^{-(I_{l2}^{h*} - \theta_{l1}^{h*})}} + \sum_{w_{l2}^{o*} < 0} \frac{w_{l2}^{o*}}{1 + e^{-(I_{l2}^{h*} - \theta_{l3}^{h*})}} + \ln\left(\frac{1}{N(a_j)} - 1\right) \right) \quad (62)$$

Theorem 3 is proved. \square

To derive the optimal values of $\tilde{\theta}^{o*}$ and $\tilde{\theta}_l^{h*}$, a two-step procedure is established in this study. First, the thresholds on the hidden-layer nodes are set to certain values. Then, the optimal value of the threshold on the output node can be derived according to Theorem 3. After repeating this process, the optimal values of these thresholds can be obtained. Accordingly, the following random search and local optimization of Algorithm is proposed, as follows (Algorithm 1):

Algorithm 1: Random search and local optimization

- Step 1. Set t (time index) to 1.
 Step 2. Set AR_{\min} (the minimum of AR so far) to a large positive value.
 Step 3. Set $\theta_{l3}^h = \theta_{l2}^{h*} + \zeta_l$ and $\theta_{l1}^h = \theta_{l2}^{h*} - \zeta_l$ for all l ; ζ_l and ζ_l are random numbers within $[0, v]$.
 Step 4. Derive the values of θ_1^{o*} and θ_3^{o*} according to Theorem 3.
 Step 5. Calculate o_{j1} and o_{j3} based on the updated network parameters.
 Step 6. Evaluate AR .
 Step 7. If $AR_{\min} > AR$, set AR_{\min} to AR and record the values of θ_{l3}^h , θ_{l1}^h , θ_1^{o*} and θ_3^{o*} .
 Step 8. $t = t + 1$.
 Step 9. If $t > T$ (the number of iterations), go to Step 10; otherwise, return to Step 3.
 Step 10. Stop.
-

4. Case Study

4.1. Background

With the advancement of wafer fabrication technologies, more and more advanced semiconductor devices (such as 3D NAND and finFETs) have emerged; however, the required cycle times have also become longer. For example, a 5 nm semiconductor device may have up to 100 mask layers, and each layer takes 0.8 to 1.5 days. To cope with this, wafer fabrication factories (wafer fabs) usually require faster equipment with patterning tools [41]. However, it is well known that a longer cycle time is associated with higher variation [42]. Therefore, forecasting the cycle time of a job becomes even more difficult.

The case of a wafer fab for making dynamic random access memory (DRAM) products is adopted to illustrate the proposed methodology. This case has been investigated by Chen [4]. In this case, the data of 120 jobs in the wafer fab have been collected. After a backward regression analysis, six factors were considered to be most influential to the cycle time of a job, as defined in Table 2. Therefore, $P = 6$.

Table 2. Factors influential to the cycle time of a job.

Variable	Definition
x_{j1}	job size (pieces)
x_{j2}	fab work-in-process (WIP; jobs)
x_{j3}	queue length before the bottleneck (jobs)
x_{j4}	queue length on the processing route (jobs)
x_{j5}	average waiting time of recently completed jobs (h)
x_{j6}	fab utilization

The average and standard deviation of the cycle times were 1229 and 208 h, respectively, showing that the collected cycle times were highly uncertain. The data of the first 80 jobs were used to build/train the model, whereas the remaining data were reserved for testing/evaluation.

4.2. Application of the Proposed Methodology

MATLAB was applied to implement the proposed methodology on a PC with an i7-7700 CPU of 3.6 GHz and 8 GB of RAM. First, to derive the cores of the network parameters, SFBPN was treated as a crisp one and was trained using the LM algorithm. To optimize the forecasting accuracy in terms of RMSE, various numbers of nodes in the hidden layer were tried. The results are summarized in Figure 3. With more than eight nodes in the hidden layer, the RMSE for fitting the training data could be reduced to a sufficiently low level (i.e., <30 h). In addition, too many nodes in the hidden layer might increase the possibility of overfitting. Therefore, the number of hidden-layer nodes was set to eight.

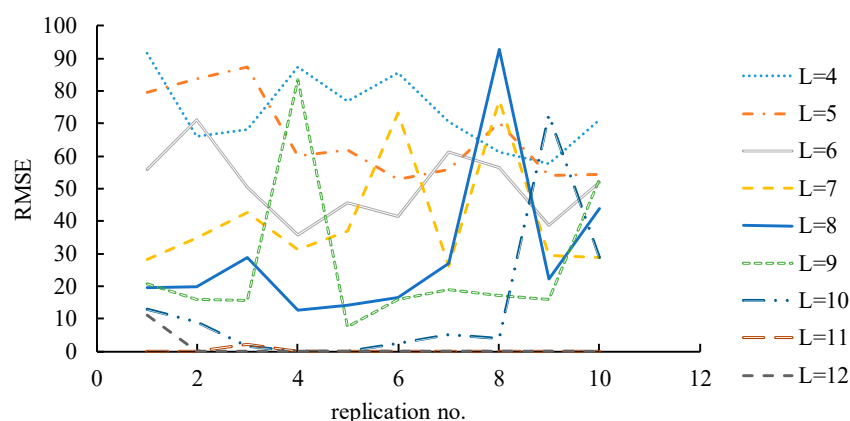


Figure 3. RMSEs associated with various numbers of hidden-layer nodes.

The trained SFBPN was then applied to generate the cores of the fuzzy cycle time forecasts. The results are shown in Figure 4. The SFBPN forecasted the cycle times accurately for the training data. However, some cycle time forecasts deviated from the actual values when the SFBPN was applied to the test data, showing the necessity for estimating the range of a cycle time.

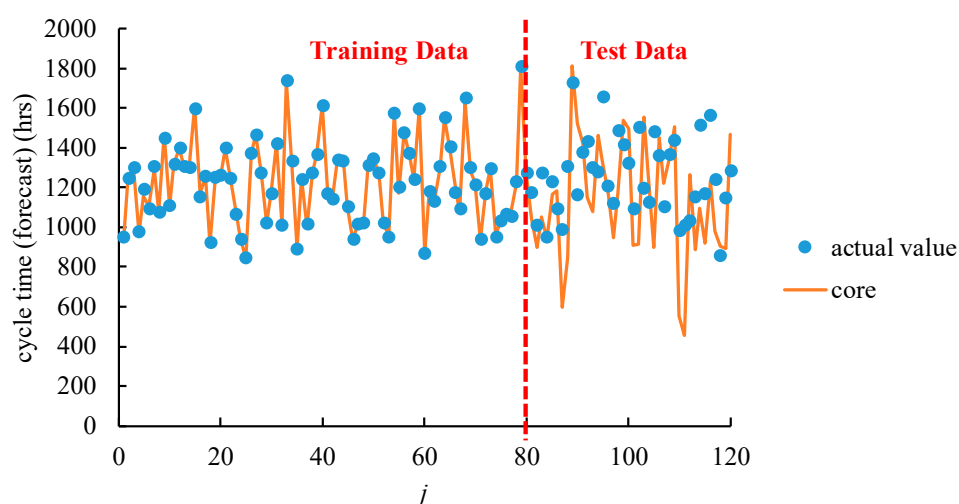
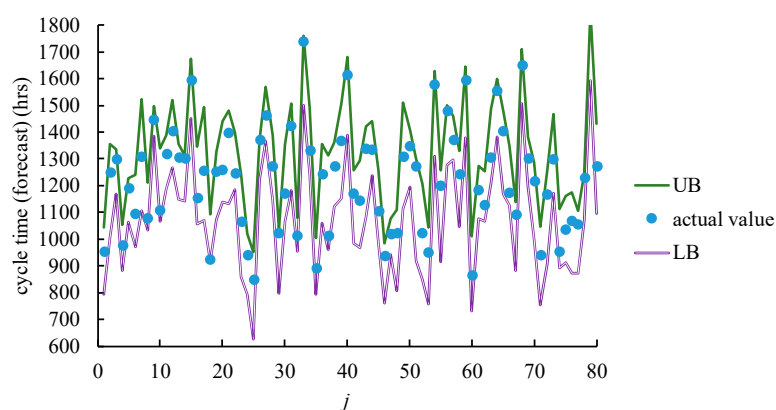
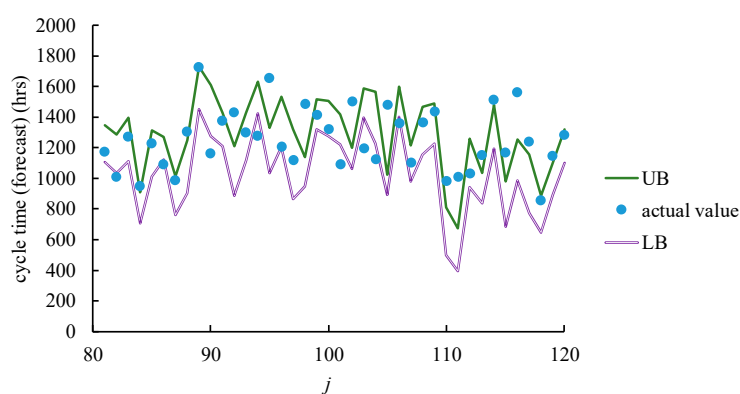


Figure 4. The cores of fuzzy cycle time forecasts.

Subsequently, the random search and local optimization algorithm was applied to estimate the range of time by fuzzifying the thresholds on the hidden and output layers. The number of iterations (T) was set to 100. The ranges of random numbers ξ_l and ζ_l were set to be within $[0, 1]$, i.e., $v = 1$. The estimation results are shown in Figure 5. For the training data, the estimated range of the cycle time contained the actual value. However, such a property might not hold for test data.



(a) Estimated ranges of cycle times (training data).



(b) Estimated ranges of cycle times (test data).

Figure 5. Estimated ranges of cycle times.

The forecasting precision was evaluated in terms of the average range (AR), hit rate (HR), and the cost for inclusion (CFI):

$$AR = \frac{\sum_{j=1}^n (U(o_{j3}) - U(o_{j1}))}{n} \quad (63)$$

$$HR = \frac{\sum_{j=1}^n In(j)}{n} \cdot 100\% \quad (64)$$

$$CFI = \frac{AR}{HR} \quad (65)$$

where

$$In(j) = \begin{cases} 1 & \text{if } U(o_{j1}) \leq a_j \leq U(o_{j3}) \\ 0 & \text{otherwise} \end{cases} \quad (66)$$

The results are summarized in Table 3.

Table 3. The forecasting precision achieved using the proposed methodology.

Data Part	AR (h)	HR	CFI (h)
Training	260	100%	260
Test	261	43%	613

4.3. Comparison with Existing Methods

Several existing methods have also been applied to this case for comparison. The first existing method is a traditional statistical analysis technique, the 6σ confidence interval method [26], in which three times the standard deviation was added to and subtracted from the core in order to determine the upper and lower bounds of the cycle time. However, in theory, the probability that a $\pm 3\sigma$ confidence interval contains an actual value is only 99.7% under the residual normality assumption. The standard deviation, σ , using the FBPN was derived as follows [1]:

$$\sigma = \sqrt{\frac{n}{n-P-1}} \cdot \text{RMSE} \quad (67)$$

In this experiment, σ was 289 h. The forecasting precision using the 6σ confidence interval method is shown in Table 4.

Table 4. The forecasting precision using the 6σ confidence interval method.

Data Part	AR (h)	HR	CFI (h)
Training	1736	100%	1736
Test	1736	100%	1736

The second existing method is the fuzzy linear regression (FLR)-quadratic programming (QP) method proposed by Donoso et al. [43], in which the relationship between the cycle time of a job and its attributes is fitted with a FLR regression, as follows:

$$\tilde{o}_j = \tilde{w}_0(+) \sum_{p=1}^P \tilde{w}_p x_{jp} \quad (68)$$

To derive the values of the fuzzy parameters in Equation (69), the following QP problem was solved:

(QP)

$$\text{Min } Z = \omega_1 \sum_{t=1}^T (o_{j2} - a_j)^2 + \omega_2 \sum_{t=1}^T (o_{j3} - o_{j1})^2 \quad (69)$$

subject to

$$a_j \geq o_{j1} + s(o_{j2} - o_{j1}) \quad 1 \leq j \leq n \quad (70)$$

$$a_j \leq o_{j3} + s(o_{j2} - o_{j3}); \quad 1 \leq j \leq n \quad (71)$$

$$o_{j1} = w_{01} + \sum_{p=1}^P w_{p1} x_{jp}; \quad 1 \leq j \leq n \quad (72)$$

$$o_{j2} = w_{02} + \sum_{p=1}^P w_{p2} x_{jp}; \quad 1 \leq j \leq n \quad (73)$$

$$o_{j3} = w_{03} + \sum_{p=1}^P w_{p3} x_{jp}; \quad 1 \leq j \leq n \quad (74)$$

$$w_{p1} \leq w_{p2} \leq w_{p3}; \quad 0 \leq p \leq P \quad (75)$$

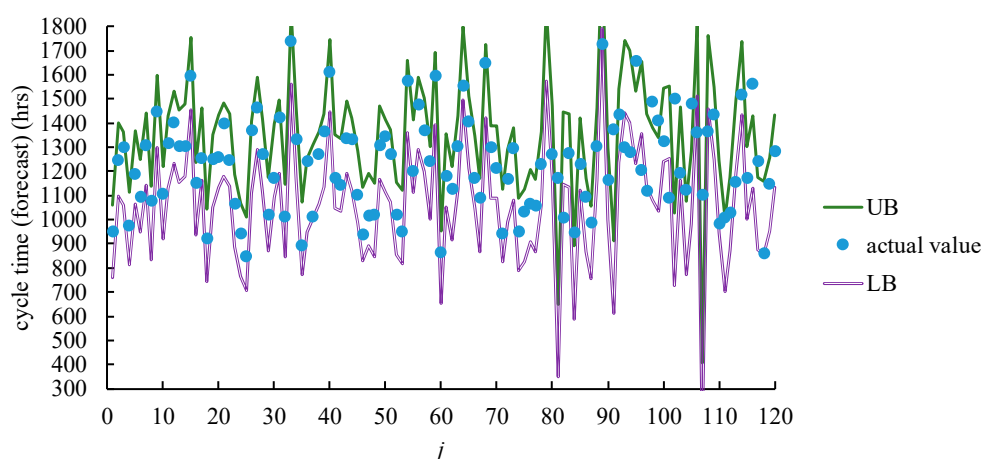
$$0 \leq o_{j1} \leq o_{j2} \leq o_{j3}; \quad 1 \leq j \leq n \quad (76)$$

where ω_1 and ω_2 are the weights; $\omega_1, \omega_2 \in [0, 1]$; $\omega_1 + \omega_2 = 1$. s is the satisfaction level; $s \in [0, 1]$. In this study, these parameters were set as $\omega_1 = 0.3$; $\omega_2 = 0.7$; $s = 0.25$. The forecasting precision achieved by applying the FLR-QP method is shown in Table 5.

Table 5. The forecasting precision achieved by applying the FLR-QP method.

Data Part	AR (h)	HR	CFI (h)
Training	617	100%	617
Test	616	100%	616

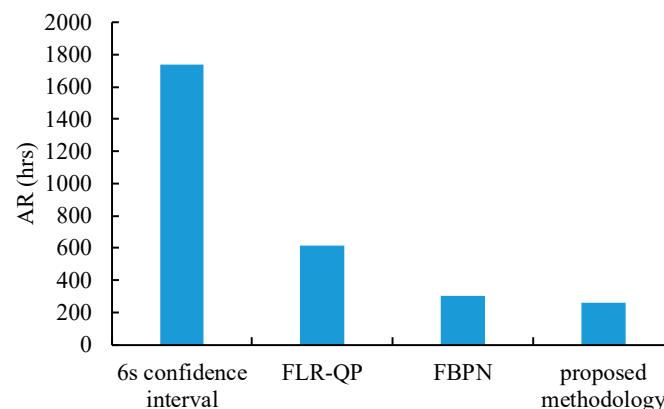
The third method is the FBPNN method proposed by Chen and Lin [17], in which only the threshold on the output node was fuzzified to estimate the range of the cycle time. The estimated ranges of cycle times are shown in Figure 6. The forecasting precision using Chen and Lin's FBPNN method was evaluated, and the results are shown in Table 6.

**Figure 6.** Estimated ranges of the cycle times using Chen and Lin's FBPNN method.**Table 6.** The forecasting precision using the FBPNN method proposed by Chen and Lin [17].

Data Part	AR (h)	HR	CFI (h)
Training	301	100%	301
Test	301	43%	709

From the experimental results, the following discussion was made:

- (1) All of the compared methods maximized the hit rate for the training data. However, the average ranges achieved using these methods differed significantly, as illustrated by Figure 7. In this regard, the proposed methodology outperformed the existing methods by establishing the narrowest range for the cycle time.

**Figure 7.** Average ranges achieved using various methods for the training data.

- (2) It is questionable whether the advantage of the SFBPN approach over the existing methods is significant. To investigate this, the following hypotheses were tested:

Hypothesis 1 (H1). *The estimation precision using the SFBPN approach in terms of the average range is the same as that using the existing method.*

Hypothesis 2 (H2). *The estimation precision using the SFBPN approach in terms of the average range is more effective than that using the existing method.*

Table 7 presents a summary of the paired t test results. The estimation precision using the SFBPN approach was significantly improved ($\alpha = 0.05$) when compared with the existing methods.

Table 7. The paired t test results.

	6 σ Confidence Interval	FLR-QP	FBPN	SFBPN
Mean	1736.5	616.8	301.1	260.0
Variation	6.4×10^{-24}	506.6	77.6	4856.5
Observations	80	80	80	80
Pearson correlation coefficient	0.005	−0.091	−0.168	
Degree of freedom	79	79	79	
t statistic	189.5	42.46	5.12	
$P(T \leq t)$ one-tail	4.4×10^{-107}	1.87×10^{-56}	1.07×10^{-6}	
t Critical one-tail	1.66	1.66	1.66	
$P(T \leq t)$ two-tail	8.7×10^{-107}	3.74×10^{-56}	2.14×10^{-6}	
t Critical two-tail	1.99	1.99	1.99	

- (3) For the test data, none of these methods optimized the hit rate and the average range simultaneously. Hit rate was usually enhanced at the expense of wide ranges of fuzzy cycle time forecasts. For this sake, CFI might be a better measure for forecasting precision. In this regard, the proposed methodology surpassed the existing methods through reducing the CFI by up to 65%, as shown in Figure 8.

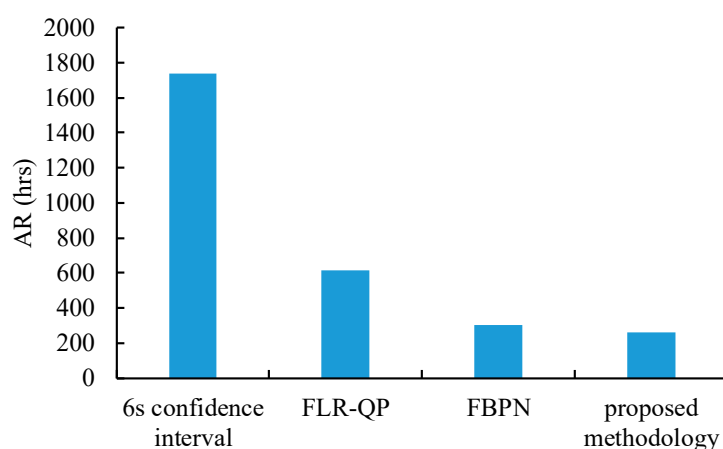


Figure 8. Cost for inclusions (CFIs) using various methods for the test data.

- (4) In the random search and local optimization algorithm, it is interesting to know whether the ranges of random numbers affected the forecasting performance of the proposed methodology. To investigate this issue, various ranges of ξ_l and ζ_l were

tried so as to observe changes in the forecasting precision. The results are summarized in Figure 9. Obviously, with a wider range, it became more difficult to find the optimal solution, which led to a poorer forecasting precision.

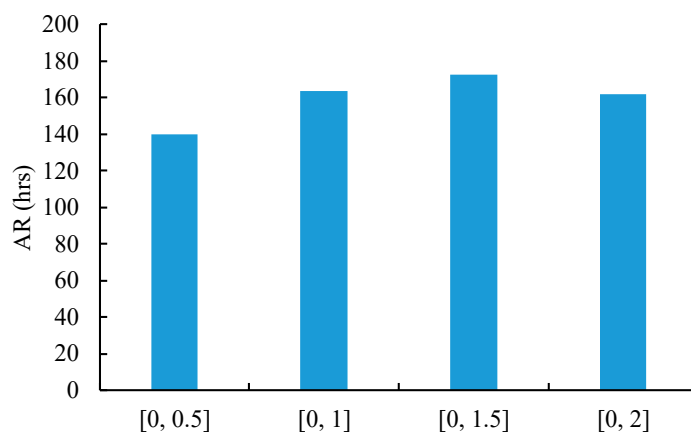


Figure 9. Effect of the range of random numbers on the average range of fuzzy cycle time forecasts.

5. Conclusions and Future Research Directions

Forecasting the cycle time of a job is a critical task for a factory. However, owing to the uncertainty of the cycle time, it becomes a challenging task, even with state-of-the-art deep learning techniques. To address this challenge, estimating the range of the cycle time requires a viable treatment. In this study, a SFBPN approach is proposed. In the proposed methodology, thresholds on both the hidden and output layers of a BPN are fuzzified to estimate the range of the cycle time, while in the existing methods, only the threshold on the output node is fuzzified. In this way, the SFBPN approach can further tighten the range of the cycle time. To derive the optimal values of the fuzzy thresholds, a random search and local optimization algorithm is also proposed.

A real case from the literature is adopted to assess the effectiveness of the proposed methodology and to compare it with those of several existing methods. According to the experimental results, the following conclusions were drawn:

- (1) For training data (i.e., learned examples), all of the compared methods were able to include the actual values in the corresponding fuzzy cycle time forecasts or cycle time confidence intervals. However, only the proposed methodology could minimize the average ranges of the fuzzy cycle time forecasts.
- (2) For the test data (i.e., unlearned examples), CFI was a better measure for the forecasting precision. In this regard, the advantage of the proposed methodology over existing methods was up to 65%.

However, the proposed methodology is subject to the following limitations:

- (1) Although the random search and local optimization algorithm is likely to find a promising solution within a short time, it cannot guarantee the global optimality of the solution.
- (2) The case used to illustrate the proposed methodology is relatively small. A larger case needs to be analyzed to further elaborate the effectiveness of the proposed methodology.

In future studies, connection weights in the SFBPN can also be fuzzified to further enhance the forecasting precision; however, this is computationally intense and requires an efficient algorithm [44–46]. As an alternative, fuzzifying some thresholds and some connection weights may be more tractable. In addition, the SFBPN approach is a general methodology that can be applied to estimate a range of data with uncertainty in other fields. Furthermore, other types of fuzzy numbers [47] can also be adopted to represent fuzzy thresholds. Hybridizing various methods to achieve total synergy [48] is another direction worth investigation.

Author Contributions: All of the authors equally contributed to the writing of this paper. Data curation, methodology, and writing original draft: T.C., Y.-C.W. and H.-R.T.; writing—review and editing: T.C. and Y.-C.W. All authors have read and agreed to the published version of the manuscript.

Funding: This study was partly funded by Ministry of Science and Technology, Taiwan.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Data is contained within the article.

Acknowledgments: This work was supported by ministry of science of technology of Taiwan.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Baykasoğlu, A.; Göçken, M.; Unutmaz, Z.D. New approaches to due date assignment in job shops. *Eur. J. Oper. Res.* **2008**, *187*, 31–45. [\[CrossRef\]](#)
2. Öztürk, A.; Kayaligil, S.; Özdemirel, N.E. Manufacturing lead time estimation using data mining. *Eur. J. Oper. Res.* **2006**, *173*, 683–700. [\[CrossRef\]](#)
3. Chang, P.C.; Liao, T.W. Combining SOM and fuzzy rule base for flow time prediction in semiconductor manufacturing factory. *Appl. Soft Comput.* **2006**, *6*, 198–206. [\[CrossRef\]](#)
4. Chen, T. An efficient and effective fuzzy collaborative intelligence approach for cycle time estimation in wafer fabrication. *Int. J. Intell. Syst.* **2015**, *30*, 620–650. [\[CrossRef\]](#)
5. Vinod, V.; Sridharan, R. Simulation modeling and analysis of due-date assignment methods and scheduling decision rules in a dynamic job shop production system. *Int. J. Prod. Econ.* **2011**, *129*, 127–146. [\[CrossRef\]](#)
6. Shabtay, D. Scheduling and due date assignment to minimize earliness, tardiness, holding, due date assignment and batch delivery costs. *Int. J. Prod. Econ.* **2010**, *123*, 235–242. [\[CrossRef\]](#)
7. Yin, Y.; Cheng, T.C.E.; Xu, D.; Wu, C.C. Common due date assignment and scheduling with a rate-modifying activity to minimize the due date, earliness, tardiness, holding, and batch delivery cost. *Comput. Ind. Eng.* **2012**, *63*, 223–234. [\[CrossRef\]](#)
8. Tirkel, I. The effectiveness of variability reduction in decreasing wafer fabrication cycle time. In Proceedings of the 2013 Winter Simulations Conference, Washington, DC, USA, 8–11 December 2013; pp. 3796–3805.
9. Asadzadeh, S.M.; Azadeh, A.; Ziaieifar, A. A neuro-fuzzy-regression algorithm for improved prediction of manufacturing lead time with machine breakdowns. *Concurr. Eng.* **2011**, *19*, 269–281. [\[CrossRef\]](#)
10. Seidel, G.; Lee, C.F.; Tang, A.Y.; Low, S.L.; Gan, B.P.; Scholl, W. Challenges associated with realization of lot level fab out forecast in a giga wafer fabrication plant. In Proceedings of the 2020 Winter Simulation Conference, Orlando, FL, USA, 14–18 December 2020; pp. 1777–1788.
11. Azadeh, A.; Ziaieifar, A.; Pichka, K.; Asadzadeh, S.M. An intelligent algorithm for optimum forecasting of manufacturing lead times in fuzzy and crisp environments. *Int. J. Logist. Manag.* **2013**, *16*, 186–210. [\[CrossRef\]](#)
12. Malik, A.I.; Sarkar, B. Coordinating supply-chain management under stochastic fuzzy environment and lead-time reduction. *Mathematics* **2019**, *7*, 480. [\[CrossRef\]](#)
13. Chen, T.; Wang, Y.C. Incorporating the FCM-BPN approach with nonlinear programming for internal due date assignment in a wafer fabrication plant. *Robot. Comput. Integr. Manuf.* **2010**, *26*, 83–91. [\[CrossRef\]](#)
14. Wang, J.; Zhang, J. Big data analytics for forecasting cycle time in semiconductor wafer fabrication system. *Int. J. Prod. Res.* **2016**, *54*, 7231–7244. [\[CrossRef\]](#)
15. Tan, X.; Xing, L.; Cai, Z.; Wang, G. Analysis of production cycle-time distribution with a big-data approach. *J. Intell. Manuf.* **2020**, *31*, 1889–1897. [\[CrossRef\]](#)
16. Chen, T.; Wu, H.C. A new cloud computing method for establishing asymmetric cycle time intervals in a wafer fabrication factory. *J. Intell. Manuf.* **2017**, *28*, 1095–1107. [\[CrossRef\]](#)
17. Chen, T.C.; Lin, Y.C. A collaborative fuzzy-neural approach for internal due date assignment in a wafer fabrication plant. *Int. J. Innov. Comput. Inf. Control.* **2011**, *7*, 5193–5210.
18. Hsieh, L.Y.; Chang, K.H.; Chien, C.F. Efficient development of cycle time response surfaces using progressive simulation metamodeling. *Int. J. Prod. Res.* **2014**, *52*, 3097–3109. [\[CrossRef\]](#)
19. Chang, P.C.; Wang, Y.W.; Liu, C.H. Combining SOM and GA-CBR for flow time prediction in semiconductor manufacturing factory. In Proceedings of the International Conference on Rough Sets and Current Trends in Computing, Kobe, Japan, 6–8 November 2006; pp. 767–775.
20. Chen, T. Asymmetric cycle time bounding in semiconductor manufacturing: An efficient and effective back-propagation-network-based method. *Oper. Res.* **2016**, *16*, 445–468. [\[CrossRef\]](#)
21. Wang, J.; Zhang, J.; Wang, X. Bilateral LSTM: A two-dimensional long short-term memory model with multiply memory units for short-term cycle time forecasting in re-entrant manufacturing systems. *IEEE Trans. Industr. Inform.* **2017**, *14*, 748–758. [\[CrossRef\]](#)
22. Goodfellow, I.; Bengio, Y.; Courville, A.; Bengio, Y. *Deep Learning*; MIT Press: Cambridge, MA, USA, 2016.

23. Montavon, G.; Samek, W.; Müller, K.R. Methods for interpreting and understanding deep neural networks. *Digit Signal Process.* **2018**, *73*, 1–15. [CrossRef]
24. Wang, J.; Zhang, J.; Wang, X. A data driven cycle time prediction with feature selection in a semiconductor wafer fabrication system. *IEEE Trans. Semicond. Manuf.* **2018**, *31*, 173–182. [CrossRef]
25. Kuo, P.H.; Huang, C.J. A high precision artificial neural networks model for short-term energy load forecasting. *Energies* **2018**, *11*, 213. [CrossRef]
26. Valles, A.; Sanchez, J.; Noriega, S.; Nuñez, B.G. Implementation of Six Sigma in a manufacturing process: A case study. *Int. J. Ind. Eng.* **2009**, *16*, 171–181.
27. Gani, A.N.; Assarudeen, S.M. A new operation on triangular fuzzy number for solving fuzzy linear programming problem. *Appl. Math. Sci.* **2012**, *6*, 525–532.
28. Wang, J.; Ding, D.; Liu, O.; Li, M. A synthetic method for knowledge management performance evaluation based on triangular fuzzy number and group support systems. *Appl. Soft Comput.* **2016**, *39*, 11–20. [CrossRef]
29. Wu, C.L.; Chau, K.W.; Fan, C. Prediction of rainfall time series using modular artificial neural networks coupled with data-preprocessing techniques. *J. Hydrol.* **2010**, *389*, 146–167. [CrossRef]
30. Chang, P.-C.; Hsieh, J.-C. A neural networks approach for due-date assignment in a wafer fabrication factory. *Eur. J. Ind. Eng.* **2003**, *10*, 55–61.
31. Martino, J.P. A review of selected recent advances in technological forecasting. *Technol. Forecast. Soc. Chang.* **2003**, *70*, 719–733. [CrossRef]
32. Tirkel, I. Forecasting flow time in semiconductor manufacturing using knowledge discovery in databases. *Int. J. Prod. Res.* **2013**, *51*, 5536–5548. [CrossRef]
33. Chiu, C.; Chang, P.C.; Chiu, N.H. A case-based expert support system for due-date assignment in a wafer fabrication factory. *J. Intell. Manuf.* **2003**, *14*, 287–296. [CrossRef]
34. Chan, K.Y.; Kwong, C.K.; Dillon, T.S.; Tsim, Y.C. Reducing overfitting in manufacturing process modeling using a backward elimination based genetic programming. *Appl. Soft Comput.* **2011**, *11*, 1648–1656. [CrossRef]
35. Chang, P.C.; Wang, Y.W.; Ting, C.J. A fuzzy neural network for the flow time estimation in a semiconductor manufacturing factory. *Int. J. Prod. Res.* **2008**, *46*, 1017–1029. [CrossRef]
36. Wang, J.; Yang, J.; Zhang, J.; Wang, X.; Zhang, W. Big data driven cycle time parallel prediction for production planning in wafer manufacturing. *Enterp. Inf. Syst.* **2018**, *12*, 714–732. [CrossRef]
37. Wang, J.; Zheng, P.; Zhang, J. Big data analytics for cycle time related feature selection in the semiconductor wafer fabrication system. *Comput. Ind. Eng.* **2020**, *143*, 106362. [CrossRef]
38. Hanss, M. *Applied Fuzzy Arithmetic*; Springer: Berlin/Heidelberg, Germany, 2005.
39. Yu, W.; Li, X. Fuzzy identification using fuzzy neural networks with stable learning algorithms. *IEEE Trans. Fuzzy Syst.* **2004**, *12*, 411–420. [CrossRef]
40. Bonnans, J.F.; Gilbert, J.C.; Lemaréchal, C.; Sagastizábal, C.A. *Numerical Optimization: Theoretical and Practical Aspects*; Springer Science & Business Media: Berlin, Germany, 2006.
41. Lapedus, M. Battling Fab Cycle Times. Available online: <https://semiengineering.com/battling-fab-cycle-times/> (accessed on 16 February 2017).
42. Hung, Y.F.; Chang, C.B. Dispatching rules using flow time predictions for semiconductor wafer fabrications. *J. Chin. Inst. Ind Eng.* **2002**, *19*, 67–75. [CrossRef]
43. Donoso, S.; Marin, N.; Vila, M.A. Quadratic programming models for fuzzy regression. In Proceedings of the International Conference on Mathematical and Statistical Modeling in Honor of Enrique Castillo, Ciudad Real, Spain, 28–30 June 2006.
44. Nauck, D.; Kruse, R. A fuzzy neural network learning fuzzy control rules and membership functions by fuzzy error backpropagation. In Proceedings of the IEEE International Conference on Neural Networks, San Francisco, CA, USA, 28 March–1 April 1993; pp. 1022–1027.
45. Ishibuchi, H.; Kwon, K.; Tanaka, H. A learning algorithm of fuzzy neural networks with triangular fuzzy weights. *Fuzzy Sets Syst.* **1995**, *71*, 277–293. [CrossRef]
46. He, W.; Dong, Y. Adaptive fuzzy neural network control for a constrained robot using impedance learning. *IEEE Trans. Neural Netw. Learn. Syst.* **2017**, *29*, 1174–1186. [CrossRef]
47. Pamucar, D.; Ecer, F.; Deveci, M. Assessment of alternative fuel vehicles for sustainable road transportation of United States using integrated fuzzy FUCOM and neutrosophic fuzzy MARCOS methodology. *Sci. Total Environ.* **2021**, *788*, 147763. [CrossRef] [PubMed]
48. Deveci, M.; Pamucar, D.; Gokasar, I. Fuzzy Power Heronian function based CoCoSo method for the advantage prioritization of autonomous vehicles in real-time traffic management. *Sustain. Cities Soc.* **2021**, *69*, 102846. [CrossRef]