A Hybrid Big Data Analytics and Industry 4.0 Approach for Projecting The Cycle Time Range

Toly Chen  
National Yang Ming Chiao Tung University Taiwan

Yu Cheng Wang  (✉ tony.cobra@msa.hinet.net)  
Chaoyang University of Technology  https://orcid.org/0000-0003-4339-1665

Research Article

Keywords: cycle time range, projection, big data analytics, Industry 4.0

Posted Date: November 8th, 2021

DOI: https://doi.org/10.21203/rs.3.rs-1031507/v1

License: © This work is licensed under a Creative Commons Attribution 4.0 International License.  
Read Full License
A hybrid big data analytics and Industry 4.0 approach for projecting the cycle time range

Toly Chen
Department of Industrial Engineering and Management
National Yang Ming Chiao Tung University
Taiwan
tolychen@ms37.hinet.net

Yu-Cheng Wang
Department of Aeronautical Engineering
Chaoyang University of Technology
Taichung City, Taiwan
tony.cobra@msa.hinet.net

Abstract: To enhance the effectiveness of projecting the cycle time range of a job in a factory, a hybrid big data analytics and Industry 4.0 (BD-I4) approach is proposed in this study. As a joint application of big data analytics and Industry 4.0, the BD-I4 approach is distinct from existing methods in this field. In the BD-I4 approach, first, each expert constructs a fuzzy deep neural network (FDNN) to project the cycle time range of a job, which is an application of big data analytics (i.e., deep learning). Subsequently, fuzzy weighted intersection (FWI) is applied to aggregate the cycle time ranges projected by experts to consider their unequal authority levels, which is an application of Industry 4.0 (i.e., artificial intelligence). After applying the BD-I4 approach to a real case, the experimental results showed that the proposed methodology improved the projection precision by up to 72%. This result implied that instead of relying on a single expert, seeking the collaboration among multiple experts may be more effective and efficient.

Keywords: cycle time range; projection; big data analytics; Industry 4.0
1. Introduction

After a job is released into the factory, it may be delayed due to a variety of reasons (e.g., waiting, machine down, product quality problems, etc.), resulting in a long cycle time [1]. These delays are often unpredictable, which leads to the uncertainty in the cycle time of a job. Nevertheless, accurate job cycle time forecasts are an input required for many managerial activities [2-5]. For this reason, various methods, such as fuzzy logic [6], artificial neural networks [6], case-based reasoning and expert systems [7], regression [6], data mining and big data analytics [8-10], production simulation [11], and hybrid methods [2, 12-14], have been proposed to predict the cycle time of a job. The experimental results of some past studies showed that it is not easy to accurately predict the cycle time of a job, especially in a large and complex factory [14-18]. Some researchers therefore tried to project the range of the cycle time of a job by constructing a confidence interval [6, 10-11, 15] or generating a fuzzy cycle time forecast [2, 12-13, 19], and then took some measures to narrow the range.

The ranges of cycle times should include actual values to maximize the hit rate and be as narrow as possible to minimize the average range. These two goals are not easy to balance. Blindly narrowing the average range often leads to a sharp decline in the hit rate. To address this issue, a hybrid big data analytics and Industry 4.0 (BD-I4) approach is proposed in this study. The BD-I4 approach combines deep learning and artificial intelligence, which are two critical subfields of big data analytics and Industry 4.0 [20-24].

In the proposed BD-I4 approach, a group of experts (or agents) collaborate to project the cycle time range of a job following the steps below. First, each expert constructs a fuzzy deep neural network (FDNN) to project the cycle time range of a job:

1. The configurations of the FDNNs constructed by different experts may not be the same: It is a challenging task to determine the optimal configuration of a FDNN. A deeper and larger FDNN increases the accuracy of cycle time forecasting for the training data, but is trained slower and may over-fit, resulting in a low hit rate for test data.

2. The FDNN is constructed by fuzzifying the parameters of a deep neural network (DNN). The parameters fuzzified by different experts may not be the same.
Subsequently, the fuzzy weighted intersection (FWI) operator [25] is adopted to aggregate the cycle time ranges projected by all experts. Over existing aggregators such as fuzzy intersection (FI), FWI has the following advantages [26]:
(1) Sometime experts have unequal authority levels, which can be considered by the FWI operator.
(2) If experts lack a consensus, the FI result is an empty set, but the FWI result may be not.

The contribution of the BD-I4 approach includes:
(1) It is the combination and application of big data analytics and Industry 4.0 to projecting the range of the cycle time. Similar treatments have rarely been taken in the past.
(2) How to construct a precise FDNN by fuzzifying a DNN is investigated theoretically in this study.
(3) The BD-I4 approach is a general method that can be applied to other fields.

The organization of this research is outlined as follows. First, a review of past studies is presented in Section 2. Then, the proposed BD-I4 approach is detailed in Section 3. To illustrate the applicability of the BD-I4 approach and make comparison with some existing methods, a real case study has been conducted. The results are summarized in Section 4. The conclusions of this research are given in Section 5.

2. Past Work

Deng et al. [27] proposed a fused FDNN network approach to classify images, in which two fuzzy neural network (FNNs) were constructed. One FNN was a shallow adaptive network-based fuzzy inference system (ANFIS), while the other was a FDNN. However, all parameters in the two FNNs were crisp, which is distinct from the BD-I4 approach proposed in this study. In addition, one FNN was shallow, while the other was deep. In contrast, all FDNNs in the BD-I4 approach are deep. Subsequently, the outputs from the two FNNs were aggregated by a crisp feedforward neural network.

Most FDNNs in the literature are of Takagi-Sugeno-Kang (TSK) type. For example, Rajurkar and Verma [28] constructed a three-layer FDNN, in which each node of the hidden layer represented a TSK fuzzy inference system (FIS). Therefore, the FDNN can be considered as an ensemble of several ANFISs.
Mudiyanselage et al. [29] constructed a FDNN to detect cancers. The FDNN was a deep ANFIS with seven layers. All parameters in the FDNN were crisp. However, the ANFIS was of TSK type, the rules in which may not be able to approximate the relationship between the cycle time of a job and its attributes. In addition, FISs or ANFISs cannot establish a range that definitely contains the actual value.

Liang et al. [30] proposed an artificial neural network ensemble approach to predict passenger demand. The FNN ensemble was composed of a (crisp) convolutional long short-term memory network (Conv-LSTM) and an ANFIS. The Conv-LSTM predicted passenger demand as a time series, while the ANFIS modelled the effects of external factors (such as weather, weekday, and day or night) on passenger demand. Similar to the previous references, all parameters in the two artificial neural networks were given in crisp values. The outputs from the two artificial neural networks were aggregated using a convolution method.

Qasem and Mohammadzadeh [31] constructed a deep learned type-2 FNN to identify a hyperchaotic system, predict chaotic time series, and predict the glucose level of a type-1 diabetes patient. The deep learned type-2 FNN was an ANFIS with seven layers. It was special because the output was mapped to an interval fuzzy number. However, the output did not necessarily contain the actual value.

The novelties of the proposed BD-I4 approach are described. First, the FDNN constructed in the BD-I4 approach is an FNN with nonlinear activation functions (i.e., sigmoid functions), which is distinct from the ANFISs constructed in the past studies. In addition, all parameters of the FDNN are fuzzy values that are to be optimized to minimize the average range of fuzzy forecasts, while guaranteeing that every fuzzy forecast contains the actual value.

3. Methodology

The procedure for the BD-I4 approach comprises the following steps:

Step 1. (Expert) Configure (or re-configure) a DNN to predict the cycle time of a job according to his/her own preference.
Step 2. (Expert) Train the DNN using the Levenberg–Marquardt (LM) algorithm [32].
Step 3. (Expert) Fuzzify the DNN to construct a FDNN based on his/her own preference.
Step 4. (Expert) Apply the FDNN to project the cycle time ranges of jobs.
Step 5. (Coordinator) Aggregate the projection results by all experts using the FWI operator.

Step 6. (Coordinator) Evaluate the projection precision.

Step 7. (Coordinator) If the projection precision is satisfactory, stop; otherwise, return to Step 1.

An activity diagram is provided in Fig. 1 to illustrate the procedure.

Fig. 1. Procedure of the BD-I4 approach.
3.1. Projecting the cycle time range of a job using a FDNN

In the BD-I4 approach, experts construct FDNNs with different configurations to project the cycle time range of a job, as illustrated in Fig. 2. These FDNNs have four layers: the input layer, two hidden layers, and the output layer. In addition, fuzzified parameters in these FDNNs may be different.

There are $M$ experts in the collaborative projection task. Inputs to the FDNNs of all experts are the same, i.e., $\{ x_{jp} \mid p = 1 \sim P; j = 1 \sim n \}$. Taking Expert #m as an example. These inputs are propagated through the FDNN as follows. First, from the input layer to the first hidden layer, the following operations are performed:
\[ f^{(1)}_{jl}(m) = \sum_{p=1}^{P} \left( \varphi^{(1)}_{jl}(m)x_p \right) \] (1)

\[ \eta^{(1)}_{jl}(m) = f^{(1)}_{jl}(m)\theta^{(1)}_{j} \] (2)

\[ f^{(1)}_{jl}(m) = \frac{1}{1 + e^{-\eta^{(1)}_{jl}(m)}} \] (3)

where \( \varphi^{(1)}_{jl}(m) \) is the connection weight between input node \( p \) and first-hidden-layer node \( l; l = 1 \sim L_m \). \( \theta^{(1)}_{j} \) is the threshold on first-hidden-layer node \( l \). \( f^{(1)}_{jl}(m) \) is the output from first-hidden-layer node \( l \). \((-)\) denotes fuzzy subtraction.

Between the two hidden layers, the following operations are performed:

\[ f^{(2)}_{jq}(m) = \sum_{l=1}^{L_m} \left( \varphi^{(2)}_{jl}(m)(\times)f^{(1)}_{jl}(m) \right) \] (4)

\[ \eta^{(2)}_{jq}(m) = f^{(2)}_{jq}(m)\theta^{(2)}_{j} \] (5)

\[ f^{(2)}_{jq}(m) = \frac{1}{1 + e^{-\eta^{(2)}_{jq}(m)}} \] (6)

\( \varphi^{(2)}_{jl}(m) \) is the connection weight between first-hidden-layer node \( l \) and second-hidden-layer node \( q \). \( \theta^{(2)}_{j} \) is the threshold on second-hidden-layer node \( q \). \( f^{(2)}_{jq}(m) \) is the output from second-hidden-layer node \( q \). \((\times)\) denotes fuzzy multiplication.

Outputs from the second hidden layer are aggregated on the output node,

\[ f^{(3)}_{j}(m) = \sum_{q=1}^{Q_m} \left( \psi^{(2)}_{jq}(m)(\times)f^{(2)}_{jq}(m) \right) \] (7)

and then is outputted as

\[ \theta^{(3)}_{j} = \frac{1}{1 + e^{-\eta^{(3)}_{j}(m)}} \] (8)

where

\[ \eta^{(3)}_{j}(m) = f^{(3)}_{j}(m)\theta^{(3)}_{j} \] (9)

\( \psi^{(2)}_{jq}(m) \) is the connection weight between second-hidden-layer node \( q \) and the output node. \( \theta^{(3)}_{j} \) is the threshold on the output node.

3.2. Training the FDNN
The training of the FDNN is composed of two stages. First, the cores of fuzzy parameters are derived by training the FDNN as a crisp DNN using the LM algorithm, so as to minimize the mean squared error (MSE). The optimal solution is indicated with
\[ \{ w_{pl2}^{h1} (m), \theta_{l2}^{h1} (m), w_{q2}^{h2} (m), \theta_{q2}^{h2} (m), w_{om}^{o}, \theta_{om}^{o} \mid p = 1 \sim P; l = 1 \sim L_m; q = 1 \sim Q_m \}. \]

Subsequently, the lower and upper bounds of fuzzy parameters are determined, so as to minimize the average range (AR(m)) of fuzzy cycle time forecasts:
\[ \text{Min } AR(m) = \frac{1}{n} \sum_{j=1}^{n} (o_{j3}(m) - o_{j1}(m)) \] (10)

However, deriving the optimal values of all fuzzy parameters at the same time is a NLP problem [13] that is computationally intensive. As an alternative, the optimal values of fuzzy parameters are derived separately as follows.

### 3.2.1. Deriving the optimal value of \( \theta^o(m) \)

\( \theta^o(m) \) is the parameter closest to the output node, fuzzifying which is the most effective way to estimate the range of the output. The core of \( \theta^o(m), \theta^o_o(m) \), has been obtained. The lower and upper bounds of \( \theta^o(m) \) can be derived according to the following theorems.

**Theorem 1.**
\[ \theta_{om}^{o} = \min_j \left( I_{j2}^{o} (m) + \ln\left( \frac{1}{\max(o_{j2}^{o} (m), a_j)} - 1 \right) \right) \] (11)

**Proof.**
Substituting Equation (9) into Equation (8) gives
\[ \delta^i(m) = \frac{1}{1 + e^{(\theta^i(m) - \delta^i(m))}} \] (12)

Therefore,
\[ o_{j3}(m) = \frac{1}{1 + e^{(\theta^o_j(m) - I_{j3}^o(m))}} \] (13)
The other parameters are not fuzzified, so $I_{j3}^o(m)$ is equal to $I_{j2}^o(m)$ that is a fixed value:

$$o_{j3}(m) = \frac{1}{1 + e^{(\theta_j^o(m) - I_{j2}^o(m))}}$$  \hspace{1cm} (14)

To minimize $AR(m)$, $o_{j3}(m)$ should be as low as possible, which corresponds to the maximization of $\theta_j^o(m)$. However, $\theta_j^o(m)$ should include $a_j$:

$$o_{j3}(m) \geq a_j; \ j = 1 \sim n$$  \hspace{1cm} (15)

In addition,

$$o_{j3}(m) \geq o_{j2}^o(m); \ j = 1 \sim n$$  \hspace{1cm} (16)

Therefore,

$$o_{j3}(m) \geq \max(o_{j2}^o(m), a_j); \ j = 1 \sim n$$  \hspace{1cm} (17)

Substituting Equation (13) into Constraint (17) gives

$$\theta_j^o(m) \leq I_{j2}^o(m) + \ln(\frac{1}{\max(o_{j2}^o(m), a_j)} - 1); \ j = 1 \sim n$$  \hspace{1cm} (18)

To maximize $\theta_j^o(m)$,

$$\theta_j^{o*}(m) = \min_j \left(I_{j2}^o(m) + \ln(\frac{1}{\max(o_{j2}^o(m), a_j)} - 1)\right); \ j = 1 \sim n$$  \hspace{1cm} (19)

Theorem 1 is proved.

**Theorem 2.**

$$\theta_j^{o*}(m) = \max_j \left(I_{j2}^o(m) + \ln(\frac{1}{\min(o_{j2}^o(m), a_j)} - 1)\right)$$  \hspace{1cm} (20)

**Proof.**

The required proof is similar to that of Theorem 1.

3.2.2. Deriving the optimal value of $W_q^o(m)$

The next closest parameter to the output node is the connection weight between a second-hidden-layer node and the output node, i.e., $W_q^o(m); \ q = 1 \sim Q_m$, which has
a considerable effect on the range of the output. Among the $Q_m$ connection weights, assuming $w_{qj}^o(m)$ is to be fuzzified. The core of $\mathcal{W}_q^o(m)$ has been derived in the first stage. Its lower and upper bounds can be determined by applying the following theorems.

**Theorem 3.**

$$w_{qj}^o(m) = \max_j \left\{ \frac{1}{\ln}\left( \frac{1}{\max(a_j, \alpha)} - 1 \right) - \sum_{q \neq q_j} (w_{qj}^o(m)h_{jqj}^2(m)) \right\}$$  \hspace{1cm} (21)

**Proof.**

According to Equation (7),

$$I_{j3}^o(m) = \max \left( \sum_{q=1}^{Q_m} (w_{qj}^o(m)h_{jqj}^2(m)), \sum_{q=1}^{Q_m} (w_{qj}^o(m)h_{jqj}^2(m)) \right)$$  \hspace{1cm} (22)

because $h_{jqj}^2(m)$ is positive, while $w_{qj}^o(m)$ may be negative. Substituting Equation (22) into Equation (13) gives

$$o_{j3}^o(m) = \frac{1}{1 + e^{\left( \theta_{2o}^o(m) - \ln\left( \frac{1}{\max(a_j, \alpha)} - 1 \right) - \sum_{q \neq q_j} (w_{qj}^o(m)h_{jqj}^2(m)) \right) \left( \sum_{q=1}^{Q_m} (w_{qj}^o(m)h_{jqj}^2(m)) \right)}}$$  \hspace{1cm} (23)

Only $w_{qj}^o$ is fuzzified, the other parameters are set to their optimized cores:

$$o_{j3}^o(m) = \frac{1}{1 + e^{\left( \theta_{2o}^o(m) - \ln\left( \frac{1}{\max(a_j, \alpha)} - 1 \right) - \sum_{q \neq q_j} (w_{qj}^o(m)h_{jqj}^2(m)) \right) \left( \sum_{q=1}^{Q_m} (w_{qj}^o(m)h_{jqj}^2(m)) \right)}}$$

$$= \frac{1}{1 + e^{\left( \theta_{2o}^o(m) - \ln\left( \frac{1}{\max(a_j, \alpha)} - 1 \right) - \sum_{q \neq q_j} (w_{qj}^o(m)h_{jqj}^2(m)) \right) \left( \sum_{q=1}^{Q_m} (w_{qj}^o(m)h_{jqj}^2(m)) \right)}}$$

(24)

Substituting Equation (24) into Constraint (17), we have

$$\frac{1}{1 + e^{\left( \theta_{2o}^o(m) - \ln\left( \frac{1}{\max(a_j, \alpha)} - 1 \right) - \sum_{q \neq q_j} (w_{qj}^o(m)h_{jqj}^2(m)) \right) \left( \sum_{q=1}^{Q_m} (w_{qj}^o(m)h_{jqj}^2(m)) \right)}} \geq \max(a_j^*, a_j) ; j = 1 \sim n$$

(25)

As a result,
\[
\theta_2^o(m) - \ln\left(\frac{1}{\max(a_j, a_j)} - 1\right) - \sum_{q\neq q_f} (w_{q_f}^o(m)h_{q_f}^{(2r)}(m) \right)
\]

\[
w_{q_j,3}^o(m) \geq \max_j \left( \frac{\theta_2^o(m) - \ln\left(\frac{1}{\max(a_j, a_j)} - 1\right) - \sum_{q\neq q_f} (w_{q_f}^o(m)h_{q_f}^{(2r)}(m))}{h_{q_j,2}^{(2r)}(m)} \right) \quad ; j = 1 \sim n
\]

Therefore,

\[
w_{q_j,3}^o(m) = \max_j \left( \frac{\theta_2^o(m) - \ln\left(\frac{1}{\max(a_j, a_j)} - 1\right) - \sum_{q\neq q_f} (w_{q_f}^o(m)h_{q_f}^{(2r)}(m))}{h_{q_j,2}^{(2r)}(m)} \right)
\]

Widening \(w_{q_j}^o(m)\) will increase the fuzziness of \(\theta_j^o(m)\). Therefore, it is reasonable to set

\[
w_{q_j,3}^o(m) = \max_j \left( \frac{\theta_2^o(m) - \ln\left(\frac{1}{\max(a_j, a_j)} - 1\right) - \sum_{q\neq q_f} (w_{q_f}^o(m)h_{q_f}^{(2r)}(m))}{h_{q_j,2}^{(2r)}(m)} \right)
\]

Theorem 3 is proved.

**Theorem 4.**

\[
w_{q_f,1}^o(m) = \min_j \left( \frac{\theta_3^o(m) - \ln\left(\frac{1}{\min(a_j, a_j)} - 1\right) - \sum_{q\neq q_f} (w_{q_f}^o(m)h_{q_f}^{(2r)}(m))}{h_{q_f,2}^{(2r)}(m)} \right)
\]

**Proof.**

The required proof is similar to that of Theorem 3.

In addition, if \(\theta^o(m)\) and \(w_{q_f}^o(m)\) are fuzzified sequentially, then the following theorem holds.

**Theorem 5.**

\[
w_{q_f,1}^o(m) = \min_j \left( \frac{\theta_3^o(m) - \ln\left(\frac{1}{\min(a_j, a_j)} - 1\right) - \sum_{q\neq q_f} (w_{q_f}^o(m)h_{q_f}^{(2r)}(m))}{h_{q_f,2}^{(2r)}(m)} \right)
\]
\[ w_{q_{j3}}^{\phi}(m) = \max_j \left( \frac{\theta_1^{\phi}(m) - \ln\left( \frac{1}{\max(o_{j2}^{\phi}(m), a_j)} - 1 \right) - \sum_{q \neq q_j} (w_{q2}^{\phi}(m) h_{j2}^{(2)^\phi}(m))}{h_{j2}^{(2)^\phi}(m)} \right) \] (31)

**Proof.**

The required proof is straightforward since the other variables in Equations (21) and (29) are not affected by the fuzzification of \( \theta^\rho(m) \).

### 3.2.3. Deriving the optimal value of \( \theta_{q_{(2)}}^{\phi}(m) \)

\( \theta_{q_{(2)}}^{\phi}(m) \) is the third closest parameter to the output node. However, the sigmoid transformation function limits the effect of \( \theta_{q_{(2)}}^{\phi}(m) \) on the range of the output. Therefore, fuzzifying \( \theta_{q_{(2)}}^{h\phi}(m) \) may not guarantee that all fuzzy forecasts contain actual values.

Among the threshold on second-hidden-layer node \( q_f \) is fuzzified. The core of \( \theta_{q_{f}}^{h(2)}(m) \) has been optimized after the training of the DNN. The lower and upper bounds of \( \theta_{q_{f}}^{h(2)}(m) \) are derived as follows.

Theorem 6.

\[
\theta_{q_{f1}}^{h(2)^\rho}(m) = \min_j \left( \frac{\ln\left( w_{q2}^{\phi}(m) \right)}{\theta_2^{\phi}(m) - \ln\left( \frac{1}{\max(o_{j2}^{\phi}(m), a_j)} - 1 \right) - \sum_{q \neq q_j} (w_{q2}^{\phi}(m) h_{j2}^{(2)^\phi}(m))} - 1 + I_{j2}^{h(2)^\rho}(m) \right)
\] (32)

\[
\theta_{q_{f3}}^{h(2)^\rho}(m) = \max_j \left( \frac{\ln\left( w_{q2}^{\phi}(m) \right)}{\theta_2^{\phi}(m) - \ln\left( \frac{1}{\min(o_{j2}^{\phi}(m), a_j)} - 1 \right) - \sum_{q \neq q_j} (w_{q2}^{\phi}(m) h_{j2}^{(2)^\phi}(m))} - 1 + I_{j2}^{h(2)^\rho}(m) \right)
\] (33)

**Proof.**

Substituting Equation (5) into Equation (6) gives
\[
K_{jq}^{(2)}(m) = \frac{1}{1 + e^{(\rho^{(2)}_{q}(m) - \rho^{(2)}_{jq}(m))}}
\]  
which is decomposed into,

\[
I_{jq}^{(2)} = \frac{1}{1 + e^{(\rho^{(2)}_{q}(m) - \rho^{(2)}_{jq}(m))}}
\]

Substituting Equations (35) and (36) into Equation (22) leads to

\[
I_{j3}^{\alpha}(m) = \max \left\{ \sum_{q=1}^{Q_{m}} \frac{w_{q}^{\alpha}(m)}{1 + e^{(\rho^{(2)}_{q}(m) - \rho^{(2)}_{jq}(m))}} \right\}
\]

In a similar way, we can derive

\[
I_{j1}^{\alpha}(m) = \min \left\{ \sum_{q=1}^{Q_{m}} \frac{w_{q}^{\alpha}(m)}{1 + e^{(\rho^{(2)}_{q}(m) - \rho^{(2)}_{jq}(m))}} \right\}
\]

Substituting Equation (37) into Equation (14), we have

\[
o_{j3}(m) = \frac{1}{1 + e^{(\rho^{(2)}_{q}(m) - \rho^{(2)}_{jq}(m))}} \max \left\{ \sum_{q=1}^{Q_{m}} \frac{w_{q}^{\alpha}(m)}{1 + e^{(\rho^{(2)}_{q}(m) - \rho^{(2)}_{jq}(m))}} \right\}
\]

In a similar way, we also have

\[
o_{j1}(m) = \frac{1}{1 + e^{(\rho^{(2)}_{q}(m) - \rho^{(2)}_{jq}(m))}} \min \left\{ \sum_{q=1}^{Q_{m}} \frac{w_{q}^{\alpha}(m)}{1 + e^{(\rho^{(2)}_{q}(m) - \rho^{(2)}_{jq}(m))}} \right\}
\]

The following requirements should be met:

\[
1 + e^{\left(\rho^{(2)}_{q}(m) - \rho^{(2)}_{jq}(m)\right)} \geq \max(o_{j2}^{\alpha}(m), a_{j}); j = 1 \sim n
\]

\[
1 + e^{\left(\rho^{(2)}_{q}(m) - \rho^{(2)}_{jq}(m)\right)} \leq \min(o_{j2}^{\alpha}(m), a_{j}); j = 1 \sim n
\]

Equivalently,
\[
\begin{align*}
\max \left( \sum_{q=1}^{Q_n} \frac{w_{q3}^o(m)}{1+e^{(\theta_{q3}^o(m)-\theta_{nj3}^{(2)}(m))}} \right) 
&\geq \theta^o(m) - \ln\left( \frac{1}{\max(o_{j2}^*(m), a_j)} - 1 \right) 
\end{align*}
\]
\[\vdots j = 1 \sim n\]
\[
\min \left( \sum_{q=1}^{Q_n} \frac{w_{q3}^o(m)}{1+e^{(\theta_{q3}^o(m)-\theta_{nj3}^{(2)}(m))}} \right) 
&\leq \theta^o(m) - \ln\left( \frac{1}{\min(o_{j2}^*(m), a_j)} - 1 \right) 
\end{align*}
\]
\[\vdots j = 1 \sim n\]

Only \(\theta_{qj}^{(2)}\) is fuzzified, the other fuzzy parameters are set to their optimized cores:

\[
\begin{align*}
\max \left( \sum_{q\neq q_j} \frac{w_{q2}^{o*}(m)}{1+e^{(\theta_{q2}^{o*}(m)-\theta_{nj2}^{(2)*}(m))}} + \frac{w_{q2}^{o*}(m)}{1+e^{(\theta_{q2}^{o*}(m)-\theta_{nj2}^{(2)*}(m))}} \right) 
&\geq \theta^o(m) - \ln\left( \frac{1}{\max(o_{j2}^*(m), a_j)} - 1 \right) 
\end{align*}
\]
\[\vdots j = 1 \sim n\]
\[
\min \left( \sum_{q\neq q_j} \frac{w_{q2}^{o*}(m)}{1+e^{(\theta_{q2}^{o*}(m)-\theta_{nj2}^{(2)*}(m))}} + \frac{w_{q2}^{o*}(m)}{1+e^{(\theta_{q2}^{o*}(m)-\theta_{nj2}^{(2)*}(m))}} \right) 
&\leq \theta^o(m) - \ln\left( \frac{1}{\min(o_{j2}^*(m), a_j)} - 1 \right) 
\end{align*}
\]
\[\vdots j = 1 \sim n\]

If \(w_{q2}^{o*}(m) \geq 0\),

\[
\begin{align*}
\max \left( \sum_{q\neq q_j} \frac{w_{q2}^{o*}(m)}{1+e^{(\theta_{q2}^{o*}(m)-\theta_{nj2}^{(2)*}(m))}} + \frac{w_{q2}^{o*}(m)}{1+e^{(\theta_{q2}^{o*}(m)-\theta_{nj2}^{(2)*}(m))}} \right) 
&\geq \theta^o(m) - \ln\left( \frac{1}{\max(o_{j2}^*(m), a_j)} - 1 \right) 
\end{align*}
\]
\[\vdots j = 1 \sim n\]
\[
\min \left( \sum_{q\neq q_j} \frac{w_{q2}^{o*}(m)}{1+e^{(\theta_{q2}^{o*}(m)-\theta_{nj2}^{(2)*}(m))}} + \frac{w_{q2}^{o*}(m)}{1+e^{(\theta_{q2}^{o*}(m)-\theta_{nj2}^{(2)*}(m))}} \right) 
&\leq \theta^o(m) - \ln\left( \frac{1}{\min(o_{j2}^*(m), a_j)} - 1 \right) 
\end{align*}
\]
\[\vdots j = 1 \sim n\]

Therefore,
\[ \theta_{q_{j1}}^{h(2)}(m) \leq \ln\left( \frac{w_{q_{j2}}^{o^*}(m)}{\theta_2^{o^*}(m) - \ln\left( \frac{1}{\max(o_{j2}^*(m), a_j)} - 1 \right) - \sum_{q \neq q_f} \frac{w_{q2}^{o^*}(m)}{1 + e^{(h_{q2}^{[2]}(m) - h_{q2}^{[2]^*}(m))}} \right) - 1 + I_{j_{q2}}^{h(2)}(m) \]
\[ j = 1 \sim n \]  
(49)

\[ \theta_{q_{j3}}^{h(2)}(m) \geq \ln\left( \frac{w_{q_{j2}}^{o^*}(m)}{\theta_2^{o^*}(m) - \ln\left( \frac{1}{\min(o_{j2}^*(m), a_j)} - 1 \right) - \sum_{q \neq q_f} \frac{w_{q2}^{o^*}(m)}{1 + e^{(h_{q2}^{[2]}(m) - h_{q2}^{[2]^*}(m))}} \right) - 1 + I_{j_{q2}}^{h(2)}(m) \]
\[ j = 1 \sim n \]  
(50)

So,

\[ \theta_{q_{j1}}^{h(2)}(m) \leq \min_j \left\{ \ln\left( \frac{w_{q_{j2}}^{o^*}(m)}{\theta_2^{o^*}(m) - \ln\left( \frac{1}{\max(o_{j2}^*(m), a_j)} - 1 \right) - \sum_{q \neq q_f} \frac{w_{q2}^{o^*}(m)}{1 + e^{(h_{q2}^{[2]}(m) - h_{q2}^{[2]^*}(m))}} \right) - 1 + I_{j_{q2}}^{h(2)}(m) \right\} \]
(51)

\[ \theta_{q_{j3}}^{h(2)}(m) \geq \max_j \left\{ \ln\left( \frac{w_{q_{j2}}^{o^*}(m)}{\theta_2^{o^*}(m) - \ln\left( \frac{1}{\min(o_{j2}^*(m), a_j)} - 1 \right) - \sum_{q \neq q_f} \frac{w_{q2}^{o^*}(m)}{1 + e^{(h_{q2}^{[2]}(m) - h_{q2}^{[2]^*}(m))}} \right) - 1 + I_{j_{q2}}^{h(2)}(m) \right\} \]
(52)

To reduce the fuzziness of \( \theta_{q_f}^{h(2)}(m) \), it is reasonable to set

\[ \theta_{q_{j1}}^{h(2)}(m) = \min_j \left\{ \ln\left( \frac{w_{q_{j2}}^{o^*}(m)}{\theta_2^{o^*}(m) - \ln\left( \frac{1}{\max(o_{j2}^*(m), a_j)} - 1 \right) - \sum_{q \neq q_f} \frac{w_{q2}^{o^*}(m)}{1 + e^{(h_{q2}^{[2]}(m) - h_{q2}^{[2]^*}(m))}} \right) - 1 + I_{j_{q2}}^{h(2)}(m) \right\} \]
(53)

\[ \theta_{q_{j3}}^{h(2)}(m) = \max_j \left\{ \ln\left( \frac{w_{q_{j2}}^{o^*}(m)}{\theta_2^{o^*}(m) - \ln\left( \frac{1}{\min(o_{j2}^*(m), a_j)} - 1 \right) - \sum_{q \neq q_f} \frac{w_{q2}^{o^*}(m)}{1 + e^{(h_{q2}^{[2]}(m) - h_{q2}^{[2]^*}(m))}} \right) - 1 + I_{j_{q2}}^{h(2)}(m) \right\} \]
(54)

Otherwise (i.e., \( w_{q_{j2}}^{o^*}(m) < 0 \)), Constraints (45) and (46) become
\[
\frac{w_{q_j}^{o}(m)}{1 + e^{(\phi_{q_j}^{h(2)}(m) - \theta_{q_j}^{h(2)}(m))}} + \sum_{q = q_f} \frac{w_{q_j}^{o}(m)}{1 + e^{(\phi_{q_j}^{h(2)}(m) - \theta_{q_j}^{h(2)}(m))}} \geq \theta_{q_j}^{h(2)}(m) - \ln\frac{1}{\max(o_{j2}^*(m), a_j)} - 1
\]
\[; j = 1 \sim n\]  
\(55\)

\[
\frac{w_{q_j}^{o}(m)}{1 + e^{(\phi_{q_j}^{h(2)}(m) - \theta_{q_j}^{h(2)}(m))}} + \sum_{q = q_f} \frac{w_{q_j}^{o}(m)}{1 + e^{(\phi_{q_j}^{h(2)}(m) - \theta_{q_j}^{h(2)}(m))}} \leq \theta_{q_j}^{h(2)}(m) - \ln\frac{1}{\min(o_{j2}^*(m), a_j)} - 1
\]
\[; j = 1 \sim n\]  
\(56\)

Therefore,
\[
\phi_{q_j}^{h(2)}(m) \geq \ln\left(\frac{w_{q_j}^{o}(m)}{\theta_{q_j}^{h(2)}(m) - \ln\frac{1}{\max(o_{j2}^*(m), a_j)} - 1} - \sum_{q = q_f} \frac{w_{q_j}^{o}(m)}{1 + e^{(\phi_{q_j}^{h(2)}(m) - \theta_{q_j}^{h(2)}(m))}}\right) + I_{j2}^{h(2)}(m)
\]
\[; j = 1 \sim n\]  
\(57\)

\[
\phi_{q_j}^{h(2)}(m) \leq \ln\left(\frac{w_{q_j}^{o}(m)}{\theta_{q_j}^{h(2)}(m) - \ln\frac{1}{\min(o_{j2}^*(m), a_j)} - 1} - \sum_{q = q_f} \frac{w_{q_j}^{o}(m)}{1 + e^{(\phi_{q_j}^{h(2)}(m) - \theta_{q_j}^{h(2)}(m))}}\right) + I_{j2}^{h(2)}(m)
\]
\[; j = 1 \sim n\]  
\(58\)

The same results are derived. Theorem 6 is proved.

Other parameters can be fuzzified in similar ways. However, these parameters are far from the output node. Fuzzifying these parameters has very little effect on the range of the output. In the proposed methodology, if an expert fuzzifies multiple parameters, these parameters are fuzzified sequentially according to their closeness to the output node.

3.3. Aggregating the projected cycle time ranges using FWI

The projected cycle time ranges by all experts are aggregated using FWI [25]:
\[
\delta_f^{(all)} = \tilde{F}WI(\{\delta_f^{(m)}\})
\]
\(59\)

with the following membership function:
\[
\mu_{\delta_f^{(all)}}(x) = \min_m \mu_{\delta_f^{(m)}}(x) + \sum_m (\omega_m - \min_k \omega_k)(\mu_{\delta_f^{(m)}}(x) - \min_k \mu_{\delta_f^{(k)}}(x))
\]
\(60\)

where \(\omega_m\) is the authority level of expert \(m\); \(\sum_{m=1}^M \omega_m = 1\). The FWI result meet the following requirements [25]:

16
Aggregating the cycle time ranges projected by experts using FWI has the following advantages over existing aggregation methods such as FI and partial-consensus FI (PCFI):

1. The FWI result exists even if experts lack an overall consensus.
2. Some experts are more authoritative than others, which can be considered by applying FWI.

4. Case Study

4.1. Background

To evaluate the effectiveness of the proposed methodology, it has been applied to a wafer fabrication factory (wafer fab) to project the cycle time ranges of jobs. Although advanced forecasting methods have been proposed in recent studies to predict the cycle time of a job in a wafer fab, the forecasting accuracy, in terms of mean absolute percentage error (MAPE), was seldom less than 10% [16, 18]. For this reason, the range of the cycle time of a job needed to be projected. Three experts (an industrial engineering manager, a production planning engineer, and a management information system engineer) applied the proposed methodology to fulfill this task in a collaborative manner.

The cycle time of a job was predicted by fitting the relationship between six parameters (job size, fab work-in-process (WIP), the queue length before bottlenecks, the queue length on the processing route, the average waiting time of recently completed jobs, and fab utilization) and the cycle time. Therefore, $P = 6$. The data of 120 jobs have been collected. Among them, the data of the first $2/3$ jobs were adopted to train the FDNN, while the remaining data were left for evaluating the forecasting performance.
4.2. Application of the proposed methodology

At first, experts configured FDNNs according to their own preferences that are summarized in Table 1. MATLAB R2017a was applied to construct these FDNNs on a PC with an i7-7700 CPU of 3.6 GHz and 8 GB of RAM.

Table 1. Configurations of the FDNNs.

| $m$ | $L_m$ | $Q_m$ | Fuzzified Parameters |
|-----|-------|-------|----------------------|
| 1   | 6     | 0     | $\beta$              |
| 2   | 3     | 2     | $\beta$, $\beta^{(2)}_1$, $\beta^{(2)}_2$ |
| 3   | 6     | 2     | $\beta$, $\omega$, $\omega_2$ |

Subsequently, each FDNN was regarded as a crisp DNN and trained using the LM algorithm to determine the cores of fuzzified parameters, for which

- the maximal number of epochs was set to 50000;
- the goal for the convergence of the DNN was established as mean squared error (MSE) $< 100$ (hrs$^2$).

The training results are shown in Fig. 3. Obviously, these DNNs predicted the cycle times of jobs very accurately for the training data. The forecasting accuracy, measured in terms of mean absolute error (MAE), $\text{MAPE}$, and root mean squared error (RMSE), is presented in Table 2. However, when the DNNs were applied to test data, the forecasting accuracy was not satisfactory, showing the need for projecting the range of the cycle time.
Subsequently, the DNNs were fuzzified to construct FDNNs. The first expert fuzzified $\theta_1$, which was able to guarantee that all actual values were contained in the corresponding fuzzy cycle time forecasts for the training data. The optimal objective function value $AR^* = 350.8$ (hrs). The projected cycle time ranges are shown in Fig. 4.
Expert #2 fuzzified three thresholds on the second hidden layer and the output layer: $\theta_b$, $\theta_1^{(2)}$, and $\theta_2^{(2)}$. The projection results are summarized in Fig. 5. The requirement that all actual values were contained in the corresponding fuzzy cycle time forecasts was met again for the training data. In addition, $AR^*$ was slightly reduced to 330.8 (hrs), implying that fuzzifying multiple parameters may improve the precision of projecting the cycle time range.

Fig. 4. Cycle time ranges projected by Expert #1.

Fig. 5. Cycle time range projection results by Expert #2.
Expert #3 fuzzified two connection weights and one threshold: $\theta^e$, $w_1$, $w_2$. All actual values were also contained in the corresponding fuzzy cycle time forecasts for the training data. However, the optimal objective function value $AR^*$ increased considerably to 768.9 (hrs), implying that fuzzifying connection weights might be ineffective. The projected cycle time ranges by this expert are shown in Fig. 6.

![Fig. 6. Projected cycle time ranges by Expert #3.](image)

The projection precision achieved by these experts are compared in Table 3. It was noteworthy that although Expert #1 optimized the forecasting accuracy in terms of MAE, MAPE, and RMSE, his projection precision was inferior to that of Expert #2. In addition, although Expert #3 achieved a poor performance in minimizing AR, he elevated the hit rate to 95% for test data. These facts showed the necessity for aggregating the projection results by all experts.

| Expert # | AR (hrs) (Training) | Hit Rate (Training) | AR (hrs) (Test) | Hit Rate (Test) |
|----------|---------------------|---------------------|----------------|----------------|
| 1        | 350.8               | 100%                | 354.7          | 73%            |
| 2        | 330.8               | 100%                | 301.7          | 75%            |
| 3        | 768.9               | 100%                | 783.0          | 95%            |

Taking Job # 60 as an example. The cycle time ranges projected by the three experts were [702.1, 1052.8], [677.5, 1044.9], and [778.1, 1523.9] (hrs), respectively. The
widths of these ranges were 350.8, 367.4, and 745.7 (hrs), respectively. These ranges were aggregated using FI into [778.1, 1044.9] (hrs) with a width of only 266.7 (hrs), which was much narrower than the original three ranges. This result supported our belief that instead of relying on a single expert that fuzzified all parameters, seeking multiple experts to fuzzify selected parameters and then collaborate may be more efficient and effective. However, experts had unequal authority levels in this collaborative forecasting task. In particular, the industrial engineer manager had a higher authority level than the other experts. For these reasons, FWI was applied instead, for which the authority levels of experts were set to 0.5, 0.3, and 0.2, respectively. Taking Job #88 as an example, the aggregation result using FWI is shown in Fig. 7.

![Fig. 7. Aggregation result using FWI for Job #88.](image)

For comparison, three common aggregators, FI, fuzzy arithmetic mean (FAM), and fuzzy weighted average (FWA), were also applied to aggregate the cycle time ranges. The results are provided in Fig. 8. Obviously, the width of the aggregation result was narrower when FI or FWI was applied.
The aggregation results for all jobs in test data are summarized in Fig. 9. The average range was 429.9 (hrs), and the hit rate was 87.5%.
4.3. Comparison with existing methods

To elaborate the effectiveness of the proposed methodology, its projection performance was compared with those of four existing methods: the $6\sigma$ confidence interval method [33], the fuzzy linear regression (FLR)-quadratic programming (QP) method [24], the FBPN method [2], and the selectively fuzzified FBPN approach [34].

In the $6\sigma$ confidence interval method, the lower bound on the cycle time was established by subtracting three times the standard error from the core, while the upper bound was established by adding the same allowance to the core. The standard error of forecasting was approximated by a function of RMSE:

$$\sigma \approx \sqrt{\frac{n}{n-V-1}} \cdot \text{RMSE}$$  \hspace{1cm} (61)

where $V$ is the number of parameters in the cycle time forecasting method. In this way, the cycle time range was always six times the standard error of forecasting. Such a range was wide though, but did not necessarily contain actual value [10]. In addition, the cycle time range determined in this way was symmetric, which might be impractical [19].

In the FLR-QP method, the cycle time of a job was predicted using a FLR function of the six factors, which support was used to project the cycle time range. In theory, there are various ways to fit a FLR equation. In the FLR-QP method, a QP problem was solved.
In the FBPN method, a FBPN was constructed and trained to project the cycle time range of a job. However, only the threshold on the output node was fuzzified to simplify the required computation.

In the selectively fuzzified FBPN method, both thresholds on the hidden and output layers of the FBPN were fuzzified, which further narrowed the cycle time range. To this end, a NLP problem was solved, with the aid of a random search and local optimization algorithm. The random search and local optimization algorithm first randomized the values of thresholds on the hidden layer, and then optimized the threshold on the output node.

The performances of projecting the cycle time range using various methods are compared in Table 5. To combine AR and hit rate, the cost for inclusion (CFI) [35] was also calculated for each method:

\[ CFI = \frac{AR}{Hit\ Rate} \]  

(45)

Table 5. Projection precision achieved using various methods.

| Method                          | AR (hrs) | Hit Rate | CFI |
|---------------------------------|----------|----------|-----|
| 6σ confidence interval method  | 1736     | 100%     | 1736|
| FLR-QP                          | 616      | 100%     | 616 |
| FBPN                            | 301      | 43%      | 700 |
| Selectively fuzzified FBPN method | 261    | 43%      | 607 |
| Proposed methodology            | 430      | 88%      | 491 |

According to experimental results, the following discussion was made:

1. The case of Expert #2 showed that a more complicated (or deeper) DNN did not necessarily result in better forecasting accuracy for test data.

2. The two traditional methods based on probability theories or statistics, 6σ confidence interval method and FLR-QP, generated very wide ranges of cycle times. In contrast, fuzzified artificial neural network methods, FBPN and selectively fuzzified FBPN method, were more effective in shrinking the average range. However, their performances in maximizing the hit rate were not satisfactory.
(3) The proposed methodology overcame the difficulty of existing fuzzified artificial neural network methods by elevating the hit rate to a sufficiently high level.

(4) In addition, the proposed methodology maximized CFI, showing its ability in elevating the hit rate without considerably increasing the average range. The advantage was up to 72% over the 6σ confidence interval method.

(5) For comparison, FI was also applied to aggregate the projected cycle time ranges by experts. The results are shown in Fig. 10. Although AR was reduced to a very low level – 213.2 (hrs), however, HR was only 57.5%. Therefore, aggregating the projected cycle time ranges using FWI seemed to be a more promising treatment.

![Fig. 9. Aggregation results using FI.](image)

4. Conclusion and Future Research Directions

The BD-I4 approach is proposed in this study to enhance the effectiveness of projecting the cycle time range of a job, which is a critical task when facing the difficulty of accurately predicting the cycle time of the job. The proposed methodology is a joint application of big data analytics and Industry 4.0, which makes it distinct from existing methods in this field. In the BD-I4 approach, first, each expert constructs a FDNN to forecast the cycle time of a job and project the cycle time range. Subsequently, the cycle time ranges projected by all experts are aggregated using FWI. Instead of relying on an expert that tries to fuzzify all parameters, seeking the collaboration among several experts is more effective and efficient.
The BD-I4 approach has been applied to a real case to evaluate its effectiveness and make comparison with several existing methods. According to the experimental results:

(1) The BD-I4 approach effectively tightened the ranges of fuzzy cycle time forecasts, while achieving a very high hit rate for test data.

(2) FWI was shown to be a more effective aggregator than FI that led to a very low hit rate. In addition, the unequal authority levels of experts could be reasonably reflected in the aggregation result using FWI.

The authority levels are subjectively assigned in FWI. In future studies, other aggregators can be devised to aggregate the cycle time ranges projected by experts with unequal authority levels. In addition, other types of deep-learning neural networks can be constructed to project the cycle time range of a job.

Declarations

Acknowledgment: This work was supported by Ministry of Science of Technology of Taiwan.

Funding: This research received no external funding.

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

Availability of data and materials: There is no original data associated with this review.

Code availability: Not applicable.

Ethics approval: Not applicable.

Consent to participate: Not applicable.

Consent for publication: Not applicable.

Authors’ contributions: Both authors contributed equally to the writing of this paper. Both authors read and approved the final manuscript. Data curation, methodology, and
writing original draft: Toly Chen. and Yu–Cheng Wang; writing—review and editing: Toly Chen and Yu-Cheng.

References

1. Dauendorffer, A., Shiozawa, T., Yoshida, K., Nagamine, N., Kamei, Y., Kawakami, S., Shimura, S., Nafus, K., Sonoda, A., & Foubert, P. (2020). Clean track solutions for defectivity and CD control towards 5 nm and smaller nodes. Extreme Ultraviolet (EUV) Lithography XI, 11323, 113232A.

2. Chen, T., Wang, Y. C., & Tsai, H. R. (2009). Lot cycle time prediction in a ramping-up semiconductor manufacturing factory with a SOM–FBPN-ensemble approach with multiple buckets and partial normalization. The International Journal of Advanced Manufacturing Technology, 42(11), 1206-1216.

3. Baykasoğlu A., Göçken, M., & Unutmaz, Z. D. (2008). New approaches to due date assignment in job shops. European Journal of Operational Research, 187(1), 31-45.

4. Shabtay, D. (2010). Scheduling and due date assignment to minimize earliness, tardiness, holding, due date assignment and batch delivery costs. Int J Prod Econ., 123(1), 235-242.

5. Yin, Y., Cheng, T. C. E., Xu, D., & Wu, C. C. (2012). 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., 63(1), 223-234.

6. Azadeh, A., Ziaeifar, A., Pichka, K., & Asadzadeh, S. M. (2013). An intelligent algorithm for optimum forecasting of manufacturing lead times in fuzzy and crisp environments. Int. J. Logist. Manag., 16(2), 186-210.

7. Chiu, C., Chang, P. C., & Chiu, N. H. (2003). A case-based expert support system for due-date assignment in a wafer fabrication factory. Journal of Intelligent Manufacturing 2003, 14(3), 287-296.

8. Öztürk, A., Kayaligil, S., & Özdemirel, N. E. (2006). Manufacturing lead time estimation using data mining. Eur. J. Oper. Res., 173(2), 683-700.

9. Wang, J., & Zhang, J. (2016). Big data analytics for forecasting cycle time in semiconductor wafer fabrication system. Int. J. Prod. Res., 54(23), 7231-7244.
10. Tan, X., Xing, L., Cai, Z., & Wang, G. (2020). Analysis of production cycle-time distribution with a big-data approach. Journal of Intelligent Manufacturing, 31, 1889–1897.

11. Vinod, V., & Sridharan, R. (2011). Simulation modeling and analysis of due-date assignment methods and scheduling decision rules in a dynamic job shop production system. Int J Prod Econ., 129(1), 127-146.

12. Chang, P. C., & Liao, T. W. (2006). Combining SOM and fuzzy rule base for flow time prediction in semiconductor manufacturing factory. Appl. Soft Comput., 6(2), 198-206.

13. Chen, T. (2012). An efficient and effective fuzzy collaborative intelligence approach for cycle time estimation in wafer fabrication. Int. J. Intell. Syst., 30(6), 620-650.

14. Asadzadeh, S. M., Azadeh, A., & Ziaeifar, A. (2011). A neuro-fuzzy-regression algorithm for improved prediction of manufacturing lead time with machine breakdowns. Concurr Eng., 19(4), 269-281.

15. Tirkel, I. (2013). The effectiveness of variability reduction in decreasing wafer fabrication cycle time. In Winter Simulations Conference, pp. 3796-3805.

16. Wang, J., Zhang, J., & Wang, X. (2017). 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 Transactions on Industrial Informatics, 14(2), 748-758.

17. Chien, C. F., Hsiao, C. W., Meng, C., Hong, K. T., & Wang, S. T. (2005). Cycle time prediction and control based on production line status and manufacturing data mining. In IEEE International Symposium on Semiconductor Manufacturing, 2005, pp. 327-330.

18. Wang, J., & Zhang, J. (2016). Big data analytics for forecasting cycle time in semiconductor wafer fabrication system. International Journal of Production Research, 54(23), 7231-7244.

19. Chen, T., & Wu, H. C. (2017). A new cloud computing method for establishing asymmetric cycle time intervals in a wafer fabrication factory. J Intell Manuf., 28(5), 1095-1107.

20. Pedrycz, W. (2008). Collaborative architectures of fuzzy modeling. IEEE World Congress on Computational Intelligence, pp. 117-139.
21. Chen, T. C. T., & Honda, K. (2020). Fuzzy Collaborative Forecasting and Clustering: Methodology, System Architecture, and Applications. Springer International Publishing.

22. Chen, T., Liao, T. W., & Yu, F. (2015). Fuzzy collaborative intelligence and systems. International Journal of Intelligent Systems, 30, 617-619.

23. Zapata, J., Vilar, R., & Ruiz, R. (2010). An adaptive-network-based fuzzy inference system for classification of welding defects. NDT & e International, 43(3), 191-199.

24. Chen, T., & Lin, Y. C. (2008). A fuzzy-neural system incorporating unequally important expert opinions for semiconductor yield forecasting. International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems, 16(01), 35-58.

25. Chen, T. C. T., Wang, Y. C., & Lin, C. W. (2020). A fuzzy collaborative forecasting approach considering experts’ unequal levels of authority. Applied Soft Computing, 94, 106455.

26. Chen, T., Wang, Y. C., & Wu, H. C. (2021). Analyzing the impact of vaccine availability on alternative supplier selection amid the COVID-19 pandemic: A cFGM-FTOPSIS-FWI approach. Healthcare, 9(1), 71.

27. Deng, Y., Ren, Z., Kong, Y., Bao, F., & Dai, Q. (2016). A hierarchical fused fuzzy deep neural network for data classification. IEEE Transactions on Fuzzy Systems, 25(4), 1006-1012.

28. Rajurkar, S., & Verma, N. K. (2017). Developing deep fuzzy network with Takagi Sugeno fuzzy inference system. 2017 IEEE International Conference on Fuzzy Systems, pp. 1-6.

29. Mudiyanselage, T. K. B., Xiao, X., Zhang, Y., & Pan, Y. (2019). Deep fuzzy neural networks for biomarker selection for accurate cancer detection. IEEE Transactions on Fuzzy Systems, 28(12), 3219-3228.

30. Liang, X., Wang, G., Min, M. R., Qi, Y., & Han, Z. (2019). A deep spatio-temporal fuzzy neural network for passenger demand prediction. Proceedings of the 2019 SIAM International Conference on Data Mining, pp. 100-108.

31. Qasem, S. N., & Mohammadzadeh, A. (2021). A deep learned type-2 fuzzy neural network: Singular value decomposition approach. Applied Soft Computing, 105, 107244.

32. Nocedal, J., & Wright, S. (2006). Numerical Optimization; Springer Science & Business Media.
33. Valles, A.; Sanchez, J.; Noriega, S.; Nuñez, B.G. Implementation of Six Sigma in a manufacturing process: A case study. International Journal of Industrial Engineering 2009, 16(3), 171-181.

34. Wang, Y. C., Tsai, H. R., & Chen, T. (2021). A selectively fuzzified back propagation network approach for precisely estimating the cycle time range in wafer fabrication. Mathematics, 9(12), 1430.

35. Chen, T. (2017). New fuzzy method for improving the precision of productivity predictions for a factory. Neural Computing and Applications, 28(11), 3507-3520.