Information Sciences 180 (2010) 4834–4864

Contents lists available at ScienceDirect

Information Sciences

journal homepage: www.elsevier.com/locate/ins

New model for system behavior prediction based on belief rule based systems

Zhi-Jie Zhou [a,b,c], Chang-Hua Hu [a], Dong-Ling Xu [c], Jian-Bo Yang [c], Dong-Hua Zhou [b,*]

[a] *High-Tech Institute of Xi'an, Xi'an, Shaanxi 710025, PR China*
[b] *Department of Automation, TNList, Tsinghua University, Beijing 100084, PR China*
[c] *Manchester Business School, The University of Manchester, Manchester M15 6PB, UK*

ARTICLE INFO

Article history:
Received 23 April 2009
Received in revised form 10 August 2010
Accepted 12 August 2010

Keywords:
Belief rule base
Forecasting
Uncertainty
Evidential reasoning
Recursive algorithms

ABSTRACT

To predict the behavior of a complex engineering system, a model can be built and trained using historical data. However, it may be difficult to obtain a complete and accurate set of data to train the model. Consequently, the model may be incapable of predicting the future behavior of the system with reasonable accuracy. On the other hand, expert knowledge of a qualitative nature and partial historical information about system behavior may be available which can be converted into a belief rule base (BRB). Based on the unique features of BRB, this paper is devoted to overcoming the above mentioned difficulty by developing a forecasting model composed of two BRBs and two recursive learning algorithms, which operate together in an integrated manner. An initially constructed forecasting model has some unknown parameters which may be manually tuned and then trained or updated using the learning algorithms once data become available. Based on expert intervention which can reflect system operation patterns, two algorithms are developed on the basis of the evidential reasoning (ER) algorithm and the recursive expectation maximization (EM) algorithm with the former used for handling judgmental outputs and the latter for processing numerical outputs, respectively. Using the proposed algorithms, the training of the forecasting model can be started as soon as there are some data available, without having to wait until a complete set of data are all collected, which is critical when the forecasting model needs to be updated in real-time within a given time limit. A numerical simulation study shows that under expert intervention, the forecasting model is flexible, can be automatically tuned to predict the behavior of a complicated system, and may be applied widely in engineering. It is demonstrated that if certain conditions are met, the proposed recursive algorithms can converge to a local optimum. A case study is also conducted to show the wide potential applications of the forecasting model.

1. Introduction

It is very important to predict the future behaviors of complex systems. For example, in nuclear power stations and missile control loops, it is vital that system behaviors that may lead to major faults can be predicted and avoided before they occur. So far, several forecasting methods have been proposed such as filter based, signal processing based and qualitative knowledge based methods [14,61].

If the mathematical or statistical model of a system is known, filter based methods such as Kalman predictor [54], strong tracking predictor [41], particle predictor [8] and fuzzy Kalman predictor [57] can be used to predict system states or

* Corresponding author. Tel.: +86 10 62783125; fax: +86 10 62786911.
E-mail address: zdh@mail.tsinghua.edu.cn (D.-H. Zhou).

0020-0255/$ - see front matter © 2010 Elsevier Inc. All rights reserved.
doi:10.1016/j.ins.2010.08.016

estimate system parameters. If complete input and output data are known, signal processing based methods such as time series analysis based methods [16,24,30,33], grey methods [18,56] and neural networks [2,3,9,15] can be used to set up forecasting models. Future values of system outputs can then be calculated using the constructed forecasting models and used to predict future system behaviors. If qualitative knowledge about a system is known, qualitative knowledge based methods can be used to predict the future behaviors of the system, which include expert systems [1,4], knowledge based methods [23], case based reasoning [25], Petri nets [53] and so on. These forecasting methods have been used in areas such as fault prediction [8], temperature prediction [6], and reliability prediction [5,28].

However, there are specific shortcomings inherent to the above mentioned forecasting methods. Filter based methods are not applicable in cases where the accurate mathematical or statistical model of a complex system is difficult to obtain. Although signal processing based methods do not need accurate mathematical models, there is no sound theory to support the identification of the structure of such a forecasting model. Qualitative knowledge based methods may lead to combinatorial explosion and inaccurate prediction if a system is very complicated. Furthermore, the above forecasting methods use either numerical or judgmental information but not both.

In order to handle hybrid information with uncertainty in complex system modeling [36,39], a generic rule base inference methodology using the evidential reasoning (RIMER) approach has been developed to establish a nonlinear relationship between antecedent attributes (or inputs of a system to be modeled) and an associated consequent (or output) [45,49]. The RIMER approach is developed on the basis of the belief rule base (BRB) and the evidential reasoning (ER) algorithm [44,45,47–49,51,52] for multiple criteria decision analysis. The ER algorithm is based on Dempster–Shafer theory of evidence [12,32] and decision theory [17,43]. Compared with traditional IF-THEN rule base [36] and fuzzy IF-THEN rule base [7,31,37,55], the RIMER approach provides a more informative and flexible scheme for knowledge representation and is capable of capturing vagueness, incompleteness, and nonlinear causal relationships. Moreover, compared with fuzzy neural networks [21], RIMER is designed to allow direct intervention of human experts in deciding the internal structure of BRB and can be used for system optimization [46]. Equipped with the Windows-based software tool, the intelligent decision system (IDS) [50], RIMER has been applied to the safety analysis of offshore systems [26,27]. Although expert knowledge can be built into BRB, it is difficult to set the parameters of BRB accurately in a manual way, especially for large-scale rule bases with hundreds or thousands of rules. As such, several methods were proposed to train BRB [46], although these methods are for offline training and BRB is only locally trained in essence. When new data become available, BRB needs to be re-trained using all data collected. Such repeated offline training is time consuming. In order to speed up the updating of BRB, which is vital for online applications, recursive algorithms for updating BRB online were developed [58,59].

BRB established so far is for modeling static systems but not for dynamic systems. Therefore, it is not appropriate for forecasting the future behavior of a system given its current states. In order to predict the future behavior of a system according to past and current information, either numerical or qualitative, it is necessary to develop recursive algorithms which can be used to train forecasting models online for dynamic systems.

In this paper, inspired by RIMER [45] and the learning algorithms developed by Yang et al. [46] and further improved by Zhou et al. [58,59], two BRBs are constructed as a forecasting model for predicting system behavior and new recursive algorithms are also proposed for online updating these forecasting model on the basis of the recursive expectation maximization (EM) algorithm [10]. Based on the observation that the behavior of a dynamic system is normally governed by certain forces or physical principles and it follows certain patterns, the new algorithms allow the use of expert knowledge about the behaviors of a system in the training processes. In the new training algorithms, system behaviors are translated into a set of constraints that the parameters of a BRB model must follow. We call this type of constraints expert intervention in the training process. This means that in the proposed recursive algorithms for updating a forecasting model, direct intervention from human experts is allowed.

The rest of this paper is organized as follows. In Section 2, the RIMER approach is briefly reviewed and a forecasting model for system behavior prediction is constructed on the basis of BRB. Section 3 proposes two recursive algorithms, one for online updating the forecasting model under judgmental outputs and the other under numerical outputs, respectively. A numerical simulation study and a case study are then presented in Section 4 to validate the proposed recursive algorithms. The paper is concluded in Section 5.

2. Forecasting models based on belief rule bases

2.1. Belief rule base of a system

A belief rule base (BRB) consists of a collection of belief rules defined as follows [45]:

$$\begin{aligned} R_k: &\text{If } \hat{x}_1 \text{ is } A_1^k \wedge \hat{x}_2 \text{ is } A_2^k \wedge \cdots \wedge \hat{x}_{M_k} \text{ is } A_{M_k}^k, \quad \text{then } \left\{\left(D_1, \beta_{1,k}^1\right), \ldots, \left(D_N, \beta_{N,k}^1\right)\right\} \\ &\text{with a rule weight } \theta_k^1 \text{ and attribute weight } \delta_{1,k}^1, \delta_{2,k}^1, \ldots, \delta_{M_k,k}^1 \end{aligned} \tag{1}$$

where $\hat{x}_1, \hat{x}_2, \ldots, \hat{x}_{M_k}$ represent the antecedent attributes in the kth rule. A_i^k $(i = 1, \ldots, M_k,\ k = 1, \ldots, L)$ is the referential value of the ith antecedent attribute $\hat{x}_i$ in the kth rule and $A_i^k \in A_i$. $A_i = \{A_{i,j},\ j = 1, \ldots, J_i\}$ is a set of referential values for the ith antecedent attribute and J_i is the number of the referential values of the ith antecedent attribute.

$\theta_k^1(\in R^+,\ k=1,\ldots,L)$ is the relative weight of the kth rule, and $\delta_{1,k}^1,\delta_{2,k}^1,\ldots,\delta_{M_k,k}^1$ are the relative weights of the M_k antecedent attributes used in the kth rule. $\beta_{j,k}^1$ $(j=1,\ldots,N,\ k=1,\ldots,L)$ is the belief degree to which the consequent D_j is believed to be true. If $\sum_{j=1}^N\beta_{j,k}^1=1$, the kth rule is said to be complete; otherwise, it is incomplete. Note that "$\wedge$" is a logical connective to represent the "AND" relationship. In addition, suppose that M is the total number of antecedent attributes used in the rule base. The BRB that is composed of the belief rule as given in Eq. (1) is named as BRB_1, and there is $\delta_i^1=\delta_{i,k}^1$ $(i=1,\ldots,M,\ k=1,\ldots,L)$.

A BRB can be established in the following four ways [42]:

(1) Extracting belief rules from expert knowledge;
(2) Extracting belief rules by examining historical data;
(3) Using the previous rule bases if available, and
(4) Random rules without any pre-knowledge.

For a complex system, prior knowledge may not be perfect, which may lead to the construction of an incomplete or even inappropriate initial BRB structure. Also, too many rules in an initial BRB may lead to over-fitting, whilst too few rules may result in under-fitting. In order to solve these problems, a realistic method was proposed to adjust the structure and parameters of BRB in an adaptive manner [60]. If an initial BRB is complete, the parameters in the BRB can be trained using the proposed offline and online learning algorithms [46,58,59].

2.2. Belief rule inference using the evidential reasoning algorithm

Let $\hat{\mathbf{x}}=[\hat{x}_1,\ldots,\hat{x}_M]^T$. An antecedent attribute $\hat{x}_i(n)$ $(i=1,\ldots,M)$, i.e., the ith input at time instant n can be equivalently transformed to a distribution over the referential values defined for the attribute using belief degrees as follows [45]:

$$S(\hat{x}_i(n))=\{(A_{i,j},\alpha_{i,j}(n)),\ j=1,\ldots,J_i\},\quad i=1,\ldots,M, \tag{2}$$

where $A_{i,j}$ is the jth referential value of the input $\hat{x}_i(n)$. $\alpha_{i,j}(n)$ is the belief degree to the referential value $A_{i,j}$ with $\alpha_{i,j}(n)\geqslant 0$ and $\sum_{j=1}^{J_i}\alpha_{i,j}(n)\leqslant 1$ $(i=1,\ldots,M)$, and J_i is the number of the referential values that are used for describing the input $\hat{x}_i(n)$. The preceding distributed assessment reads that the input $\hat{x}_i(n)$ is assessed to the referential value $A_{i,j}$ with the belief degree $\alpha_{i,j}(n)$.

The belief degree $\alpha_{i,j}(n)$ could be generated using various ways, depending on the nature of input (for example, input can be qualitative or quantitative). Input information could be one of the following types: continuous, discrete, symbolic and ordered symbolic. In order to facilitate data collection, a scheme for handling various types of input information was summarized for the following case [44–46]:

(1) Quantitative inputs assessed using referential terms;
   (a) *Transformation based on fuzzy membership function.*
   In this case, the referential value $A_{i,j}$ ($i=1,\ldots,M, j=1,\ldots,J_i$) is fuzzy and the belief degree $\alpha_{i,j}$ can be calculated via membership functions.
   (b) *Rule- or utility-based equivalence transformation techniques for quantitative data.*
   The rule based equivalence transformation technique that will be used in the following sections of this paper is given in detail in Appendix A.
(2) Quantitative inputs assessed using interval;
In this case, the interval can be seen as a special form of the fuzzy linguistic value, so the belief degree can be determined in a way similar to the case where the referential value is fuzzy.
(3) Qualitative inputs assessed using subjective judgments;
In this case, the belief degree $\alpha_{i,j}$ is assigned directly by the decision maker using his subjective judgments for each referential value $A_{i,j}$. For example, if $\varepsilon_{i,j}$ is the belief degree assigned to $A_{i,j}$, then $\alpha_{i,j}=\varepsilon_{i,j}$.
(4) Symbolic inputs that are assessed using subjective judgments;
In this case, similar to a qualitative linguistic value, if $\varepsilon_{i,j}$ is the belief degree assigned to the symbolic term $A_{i,j}$ by the decision maker, then $\alpha_{i,j}=\varepsilon_{i,j}$.

When the input vector of BRB_1, $\hat{\mathbf{x}}(n)$, is available, the following two steps are included in the belief rule inference using the evidential reasoning (ER) algorithm [45].

Step 1: Calculation of the activation weight;

It is assumed that $\alpha_i^k(n)$ is the belief degree of the ith input $\hat{x}_i(n)$ to the referential value in the kth rule at time instant n. Here it is called individual matching degree and there is $\alpha_i^k(n)\in\{\alpha_{i,j}(n),\ i=1,\ldots,M,\ j=1,\ldots,J_i\}$.

Thus, when the kth rule is activated, the degree of activation of the kth rule at time instant $n,\omega_k^1(n)$, is calculated by [45]

$$\omega_k^1(n) = \frac{\theta_k^1 \prod_{i=1}^M \left(\alpha_i^k(n)\right)^{\bar{\delta}_i^1}}{\sum_{l=1}^L \theta_l^1 \prod_{i=1}^M \left(\alpha_i^l(n)\right)^{\bar{\delta}_i^1}} \quad \text{and} \quad \bar{\delta}_i^1 = \frac{\delta_i^1}{\max_{i=1,\ldots,M}\{\delta_i^1\}}, \tag{3}$$

where δ_i^1 $(\in R^+,\ i = 1,\ldots,M)$ is the relative weight of $\hat{x}_i(n)$ used in the kth rule. The constraints on the parameters of BRB are given in [45,46], and will be listed in Section 3.3. $\alpha_k(n) = \prod_{i=1}^M (\alpha_i^k(n))^{\bar{\delta}_i}$ is the normalized combined matching degree.

Step 2: Rule inference using the evidential reasoning approach.

When the kth rule as given in Eq. (1) is activated, i.e., $\omega_k^1(n) \neq 0$, the belief degrees $\beta_{j,k}^1$ $(j = 1,\ldots,N,\ k = 1,\ldots,L)$ are firstly transformed into basic probability masses as follows:

$$m_{j,k}(n) = \omega_k^1(n)\beta_{j,k}^1, \tag{4}$$

$$m_{D,k}(n) = 1 - \omega_k^1(n)\sum_{j=1}^N \beta_{j,k}^1, \tag{5}$$

where $m_{j,k}(n)$ denotes the probability mass assigned to consequent D_j. $m_{D,k}(n)$ denotes the probability mass which is unassigned to any individual consequent.

Then, aggregate the L rules to generate the combined belief degree in each consequent D_j $(j = 1,\ldots,N)$. Suppose $m_{j,I(k)}(n)$ is the combined probability mass in D_j by aggregating the first k rules using the Dempster's rule of combination, and $m_{D,I(k)}(n)$ is the remaining probability mass unassigned to any consequent. Let $m_{j,I(1)}(n) = m_{j,1}(n)$ and $m_{D,I(1)}(n) = m_{D,1}(n)$. Then the recursive evidential reasoning (ER) algorithm can be used to combine the first k rules as follows:

$$m_{j,I(k+1)}(n) = K_{I(k+1)}(n)\left[m_{j,I(k)}(n)m_{j,k+1}(n) + m_{j,I(k)}(n)m_{D,k+1}(n) + m_{D,I(k)}(n)m_{j,k+1}(n)\right], \tag{6}$$

$$m_{D,I(k)}(n) = \bar{m}_{D,I(k)}(n) + \tilde{m}_{D,I(k)}(n), \tag{7}$$

$$\tilde{m}_{D,I(k+1)}(n) = K_{I(k+1)}(n)\left[\tilde{m}_{D,I(k)}(n)\tilde{m}_{D,k+1}(n) + \tilde{m}_{D,I(k)}(n)\bar{m}_{D,k+1}(n) + \bar{m}_{D,I(k)}(n)\tilde{m}_{D,k+1}(n)\right], \tag{8}$$

$$\bar{m}_{D,I(k+1)}(n) = K_{I(k+1)}(n)\left[\bar{m}_{D,I(k)}(n)\bar{m}_{D,k+1}(n)\right], \tag{9}$$

$$K_{I(k+1)}(n) = \left[1 - \sum_{j=1}^N \sum_{\substack{t=1 \\ t\neq j}}^N m_{j,I(k)}(n)m_{t,k+1}(n)\right]^{-1}, \quad k = 1,\ldots,L-1, \tag{10}$$

$$\beta_j^1(n) = \frac{m_{j,I(L)}(n)}{1 - \bar{m}_{D,I(L)}(n)}, \quad j = 1,\ldots,N, \tag{11}$$

$$\beta_D^1(n) = \frac{\tilde{m}_{D,I(L)}(n)}{1 - \bar{m}_{D,I(L)}(n)}, \tag{12}$$

where $\beta_j^1(n)$ represents the belief degree assigned to D_j at time instant n. $\beta_D^1(n)$ represents the remaining belief degree unassigned to any individual D_j.

Based on the above recursive ER algorithm [44,51], the analytical ER algorithm was also proposed [40], which is proven to be equivalent to the recursive ER approach. Using the analytical ER algorithm, the final conclusion $O(\mathbf{Y}^1(n))$ is generated by aggregating all rules that are activated by the actual input vector $\hat{\mathbf{x}}(n)$. It can be represented as follows:

$$O(\mathbf{Y}^1(n)) = h(\hat{\mathbf{x}}(n)) = \left\{\left(D_j, \beta_j^1(n)\right),\quad j = 1,\ldots,N\right\}, \tag{13}$$

where $\beta_j^1(n)$ denotes the belief degree in D_j at time instant n, and is calculated as follows:

$$\beta_j^1(n) = \frac{\mu^1(n)\times\left[\prod_{k=1}^L\left(\omega_k^1(n)\beta_{j,k}^1 + 1 - \omega_k^1(n)\sum_{i=1}^N\beta_{i,k}^1\right) - \prod_{k=1}^L\left(1 - \omega_k^1(n)\sum_{i=1}^N\beta_{i,k}^1\right)\right]}{1 - \mu^1(n)\times\left[\prod_{k=1}^L\left(1 - \omega_k^1(n)\right)\right]}, \tag{14}$$

$$\mu^1(n) = \left[\sum_{j=1}^N \prod_{k=1}^L \left(\omega_k^1(n)\beta_{j,k}^1 + 1 - \omega_k^1(n)\sum_{i=1}^N \beta_{i,k}^1\right) - (N-1)\prod_{k=1}^L\left(1 - \omega_k^1(n)\sum_{i=1}^N\beta_{i,k}^1\right)\right]^{-1}, \tag{15}$$

where $\omega_k^1(n)$ is calculated by Eq. (3).

The logic behind Eq. (14) is that if the consequent in the kth rule includes D_i with $\beta_{i,k}^1 > 0$ and the kth rule is activated, the overall output must be D_i to a certain degree. The degree is measured by both the weight of the kth rule and the degree to which the antecedents of the kth rule are activated by the actual input $\hat{\mathbf{x}}(n)$ [45,46].

2.3. Forecasting model of system behavior

As given in BRB_1, system behavior at current time instant n is represented by $\mathbf{Y}^1(n)$ or $\beta_j^1(n)$ $(j = 1, \ldots, N)$. The objective of the paper is to develop a forecasting model so that its behavior at future time instant $(n + p)$ can be predicted from its behavior at current time instant n. For complex systems, we normally do not have complete or accurate data beforehand to train the model for the prediction. However, expert knowledge may be available, such as qualitative judgments and partial historical information about system behavior, which can be converted into belief rules. In order to capture the relationship between system behaviors at current time instant n and future time instant $(n + p)$, another BRB system, named BRB_2, can be constructed as follows:

$$\begin{aligned} &R_l' : \text{If } \mathbf{Y}^1(n) \text{ is } D_i, \quad \text{then } \left\{ \left(D_1, \beta_{1,l}^2(n+p)\right), \ldots, \left(D_N, \beta_{N,l}^2(n+p)\right) \right\} \\ &\text{with a rule weight } \theta_l^2 \text{ and attribute weight } \delta_{1,l}^2 \end{aligned} \tag{16}$$

where R_l' denotes the lth $(l = 1, \ldots, N)$ belief rule of BRB_2. θ_l^2 is the rule weight of the lth rule and $D_i \in \{D_1, \ldots, D_N\}$. $\mathbf{Y}^1(n)$ is the input of BRB_2 and determined by Eq. (13). $\beta_{s,l}^2(n+p)$ $(s = 1, \ldots, N)$ is the belief degree assessed to D_s after p steps and p is the forecasting step. As there is only one input in BRB_2, the weight of the attribute, denoted by $\delta_{1,l}^2$, is always 1 [45].

Eq. (13) shows that when $\mathbf{Y}^1(n)$ is available, it can be transformed and represented in terms of the referential values $D_1, \ldots, D_N$ with the belief degrees $\beta_1^1(n), \ldots, \beta_N^1(n)$. On the other hand, $D_1, \ldots, D_N$ are the referential values of the input $\mathbf{Y}^1(n)$ in BRB_2. Moreover, the ER algorithm is a general algorithm, so it can also be used for the inference of BRB_2. Using the analytical ER algorithm, the final conclusion $O(\mathbf{Y}^2(n + p))$ is generated by aggregating all rules of BRB_2 that are activated by the input $\mathbf{Y}^1(n)$. It is represented as follows:

$$O(\mathbf{Y}^2(n+p)) = g(\mathbf{Y}^1(n)) = g\left(D_j, \beta_j^1(n)\right) = \left\{ \left(D_j, \beta_j^2(n+p)\right), \ j = 1, \ldots, N \right\}, \tag{17}$$

where g denotes the analytical ER algorithm and $\mathbf{Y}^1(n)$ can be determined by Eq. (13).

From Eqs. (13) and (17), the forecasting model can be written as

$$O(\mathbf{Y}^2(n+p)) = F(\hat{\mathbf{x}}(n)) = \left\{ \left(D_j, \beta_j^2(n+p)\right), \ j = 1, \ldots, N \right\}, \tag{18}$$

where F consists of the functions h and g, and represents the relationship between the input vector (the antecedent attributes) $\hat{\mathbf{x}}(n)$ and the output of the BRB (the consequent attribute) $\beta_j^2(n+p)$ after p steps.

In the above forecasting model, there are some parameters that need to be trained or updated by the dedicated learning algorithms as discussed in the next Section, or manually adjusted so that the future behavior of the system can be predicted when $\hat{\mathbf{x}}(n)$ is given. Note that the forecasting model is composed of two BRBs, so the parameters in both BRBs will be updated simultaneously.

3. Recursive algorithms for online updating the forecasting model

In this Section, the recursive algorithms for updating the forecasting model will be described. In the development of the algorithms, it is recognized that belief is represented as probability in the ER algorithm [11,12,29,32,34,35] and it is assumed that the outputs of a forecasting model, which is composed of two BRBs, are random and independent. The independence assumption allows the use of the recursive expectation and maximization (EM) algorithm [10,13] for the development of the recursive algorithms.

In the proposed recursive algorithms, observations on system inputs and outputs are required. Similarly, we assume that a set of observation pairs $(\hat{\mathbf{x}}(n), \hat{\mathbf{y}}(n+p))$ is available, where $\hat{\mathbf{x}}(n)$ is a given input vector at current time instant and $\hat{\mathbf{y}}(n+p)$ is the corresponding observed output vector after p steps, either provided by experts or measured using instruments. $\mathbf{y}$ is the output generated by the forecasting model. Because the optimization objective is only dependent on $\hat{\mathbf{y}}$ in the proposed algorithms and $\hat{\mathbf{y}}$ can either be judgmental or numerical, we will discuss the two cases separately.

3.1. Recursive algorithm for updating forecasting model based on judgmental output

3.1.1. Recursive algorithm under judgmental output

In this case, $\hat{\mathbf{y}}(n+p)$ is judgmental and can be represented using a distributed assessment with different degrees of belief as follows:

$$\hat{\mathbf{y}}(n+p) = \left\{ \left(D_j, \hat{\beta}_j(n+p)\right), \ j = 1, \ldots, N \right\}, \tag{19}$$

where D_j is a referential (linguistic) term in the consequent part of a rule. $\hat{\beta}_j(n+p)$ is the belief degree to which D_j is confirmed by the observed data. This is the default output format in RIMER, which provides a panoramic view about output profile. This format is useful to describe judgmental output in a natural way [45,46].

Let $\widehat{\mathbf{B}} = \left[\hat{\beta}_1, \ldots, \hat{\beta}_N\right]^T$. As mentioned above, it is assumed that $\hat{\beta}_j$ $(j = 1, \ldots, N)$ is a random variable and $\widehat{\mathbf{B}}(1+p), \ldots, \widehat{\mathbf{B}}(n+p)$ are independent. Thus, the following likelihood function is obtained

$$f\left(\widehat{\mathbf{B}}(p+1),\ldots,\widehat{\mathbf{B}}(n+p)\middle|\mathbf{Q}\right)=\prod_{\tau=1}^{n}f\left(\widehat{\mathbf{B}}(\tau+p)\middle|\mathbf{Q}\right), \tag{20}$$

where $f\left(\widehat{\mathbf{B}}|\mathbf{Q}\right)$ is the conditional probability density function (pdf) of $\widehat{\mathbf{B}}$ and $\mathbf{Q}$ is the unknown parameter vector.

By Eq. (20), the expectation of the log-likelihood is defined as

$$L_{n+1}(\mathbf{Q})=E\left\{\sum_{\tau=1}^{n}\log f\left(\widehat{\mathbf{B}}(\tau+p)\middle|\mathbf{Q}\right)\middle|\mathbf{Q}(n+p)\right\}, \tag{21}$$

where $E\{\cdot|\cdot\}$ denotes the conditional expectation at $\mathbf{Q}=\mathbf{Q}(n+p)$.

From the recursive formulation, Eq. (21) can be written as

$$L_{n+1}(\mathbf{Q})=L_n(\mathbf{Q})+E\left\{\log f\left(\widehat{\mathbf{B}}(n+p)\middle|\mathbf{Q}\right)\middle|\mathbf{Q}(n+p)\right\}. \tag{22}$$

Define

$$\Gamma_1(\mathbf{Q}(n+p))\triangleq\nabla_{\mathbf{Q}}\log f\left(\widehat{\mathbf{B}}(n+p)\middle|\mathbf{Q}(n+p)\right), \tag{23}$$

$$\Xi_1(\mathbf{Q}(n+p))\triangleq E\left\{-\nabla_{\mathbf{Q}}\nabla_{\mathbf{Q}}^T\log f(\widehat{\mathbf{B}}(n+p)\middle|\mathbf{Q})\middle|\mathbf{Q}(n+p)\right\}, \tag{24}$$

where $\nabla_{\mathbf{Q}}$ is a column gradient operator with respect to $\mathbf{Q}$.

Based on the recursive EM algorithm [10,38], when Eq. (22) is maximized, $\mathbf{Q}(n+p+1)$ can be estimated as follows:

$$\mathbf{Q}(n+p+1)=\mathbf{Q}(n+p)+\frac{1}{n}[\Xi_1(\mathbf{Q}(n+p))]^{-1}\Gamma_1(\mathbf{Q}(n+p)), \tag{25}$$

where $\mathbf{Q}$ consists of the belief degrees, the rule weights, the attribute weights and other possible parameters of the two BRBs as given later.

3.1.2. Recursive algorithm under judgmental output and normal distribution assumption

In order to use the recursive algorithm as given in Eq. (25), the pdf of $\widehat{\mathbf{B}}$ needs to be given. For simplicity, the observation is assumed to obey a normal distribution in the EM algorithm [10]. In the following discussion, based on the analysis of the rationality of the assumption that $\widehat{\mathbf{B}}$ obeys a normal distribution, we will develop a new recursive algorithm for updating the forecasting model for judgmental output under the normal distribution assumption of observations.

A judgmental conclusion generated by aggregating the activated rules can also be represented using the same referential terms as for the observed output $\hat{\mathbf{y}}(n+p)$ as follows:

$$\mathbf{y}(n+p)=\{(D_j,\beta_j(n+p)),\ j=1,\ldots,N\}, \tag{26}$$

where the forecasting value, $\beta_j(n+p)$, is generated using Eq. (14) for a given input $\hat{\mathbf{x}}(n)$. According to Eq. (18), there is $\beta_j(n+p)=\beta_j^2(n+p)$ at time instant $(n+p)$.

It is desirable that the forecasting model, composed of BRB_1 and BRB_2, is updated to minimize the difference between the observed belief $\hat{\beta}_j(n+p)$ and the belief $\beta_j(n+p)$ generated by the forecasting model for each referential term. Because belief degree is adopted in a rule as given in Eq. (1) and an expert can provide continuous distributed assessments for a system state, $\hat{\beta}_j(n+p)$ can be considered as a continuous random variable and $\beta_j(n+p)$ as its expectation. Let $\mathbf{B}=[\beta_1,\ldots,\beta_N]^T$. Thus, without loss of generality, suppose that $\widehat{\mathbf{B}}(n+p)$ obeys the following complex normal distribution

$$f\left(\widehat{\mathbf{B}}(n+p)\middle|\mathbf{Q}\right)=(2\pi)^{-N/2}|\boldsymbol{\chi}|^{-1/2}\exp\left\{-\frac{1}{2}\left(\widehat{\mathbf{B}}(n+p)-\mathbf{B}(n+p)\right)^T\boldsymbol{\chi}^{-1}\left(\widehat{\mathbf{B}}(n+p)-\mathbf{B}(n+p)\right)\right\}, \tag{27}$$

where $\mathbf{Q}$ is composed of the parameter vector $\mathbf{V}=\left[\mathbf{V}_1^T,\mathbf{V}_2^T\right]^T$ and the entries of the symmetric positive definite covariance matrix $\boldsymbol{\chi}$. $\mathbf{V}_1=\left[\theta_k^1,\bar{\delta}_m^1,\beta_{j,k}^1\right]^T$ and $\mathbf{V}_2=\left[\theta_l^2,\beta_{i,l}^2\right]^T$ are the parameter vectors of BRB_1 and BRB_2, respectively, and $k=1,\ldots,L$, $m=1,\ldots,M$, $j=1,\ldots,N$, $l=1,\ldots,N$, $i=1,\ldots,N$. It is obvious that $\mathbf{V}$ is included in $\mathbf{B}(n+p)$.

In order to simplify the calculation, we assume that the elements of $\mathbf{V}$ are independent on the entries of $\boldsymbol{\chi}$. Hence, $\Gamma_1(\mathbf{Q}(n+p))$ and $\Xi_1(\mathbf{Q}(n+p))$ in Eq. (25) can be written as

$$\Gamma_1(\mathbf{Q}(n+p))=\left[\Gamma_1'(\mathbf{Q}(n+p))^T,\tilde{\Gamma}_1(\mathbf{Q}(n+p))^T\right]^T, \tag{28}$$

$$\Xi_1(\mathbf{Q}(n+p))=\begin{bmatrix}\Xi_1'(\mathbf{Q}(n+p)) & \mathbf{0}\\ \mathbf{0} & \tilde{\Xi}_1(\mathbf{Q}(n+p))\end{bmatrix}, \tag{29}$$

where $\Gamma_1'(\mathbf{Q}(n+p))$ and $\Xi_1'(\mathbf{Q}(n+p))$ are the derivatives with respect to $\mathbf{V}$. $\tilde{\Gamma}_1(\mathbf{Q}(n+p))$ and $\tilde{\Xi}_1(\mathbf{Q}(n+p))$ are the derivatives with respect to the entries of $\boldsymbol{\chi}$. Then there is

$$[\Xi_1(\mathbf{Q}(n+p))]^{-1} = \begin{bmatrix} [\Xi_1'(\mathbf{Q}(n+p))]^{-1} & \mathbf{0} \\ \mathbf{0} & \left[\tilde{\Xi}_1(\mathbf{Q}(n+p))\right]^{-1} \end{bmatrix}, \tag{30}$$

where $[\Xi_1(\mathbf{Q}(n+p))]^{-1}$ exists in any case if the inverse is interpreted as the pseudo-inverse [20].

In the following algorithm, two steps are used to estimate $\mathbf{V}$ and $\boldsymbol{\chi}$, respectively.

Step 1: When we only consider the parameter vector $\mathbf{V}$, from Eqs. (25), (28) and (30), the following equation can be obtained

$$\mathbf{V}(n+p+1) = \mathbf{V}(n+p) + \frac{1}{n}\left[\Xi_1'(\mathbf{Q}(n+p))\right]^{-1}\Gamma_1'(\mathbf{Q}(n+p)). \tag{31}$$

In Eq. (31), $\mathbf{V}(n+p)$ is known. From Eqs. (23), (24) and (27), the ath element of the gradient vector $\Gamma_1'(\mathbf{Q}(n+p))$ and the entries of $\Xi_1'(\mathbf{Q}(n+p))$ are given by

$$[\Gamma_1'(\mathbf{Q}(n+p))]_a = \frac{\partial \mathbf{B}(n+p)^T}{\partial V_a}\boldsymbol{\chi}(n+p)^{-1}\left(\widehat{\mathbf{B}}(n+p) - \mathbf{B}(n+p)\right)\Big|_{\mathbf{V}=\mathbf{V}(n+p)}, \tag{32}$$

$$[\Xi_1'(\mathbf{Q}(n+p))]_{a,b} = \frac{\partial \mathbf{B}(n+p)^T}{\partial V_a}\boldsymbol{\chi}(n+p)^{-1}\frac{\partial \mathbf{B}(n+p)}{\partial V_b}\Big|_{\mathbf{V}=\mathbf{V}(n+p)}, \tag{33}$$

where $a = 1,\ldots,L + M + L \times N + N + N \times N$ and $b = 1,\ldots,L + M + L \times N + N + N \times N$. The derivatives in Eqs. (32) and (33) are given in Appendix B.

Step 2: From Eqs. (32) and (33), it can be seen that the covariance matrix $\boldsymbol{\chi}(n+p)$ is required in order to estimate $\mathbf{V}(n+p+1)$. The algorithm to estimate $\boldsymbol{\chi}(n+p)$ is given as follows.

Since the belief degrees $\hat{\beta}_1(n+p),\ldots,\hat{\beta}_N(n+p)$ must satisfy the inequality constraint $\sum_{j=1}^N \hat{\beta}_j(n+p) \leqslant 1$, they are not independent. In order to simplify the calculation, without loss of generality, it is assumed that $\boldsymbol{\chi} = (a_{i,j})_{N\times N}$ satisfies

$$\begin{cases} a_{i,j} = \sigma_1, & i = j, \\ a_{i,j} = \sigma_2, & i \neq j. \end{cases} \tag{34}$$

Under the above assumption, there exists $\mathbf{Q} = [\mathbf{V}^T, \sigma_1, \sigma_2]^T$.

If $\hat{\mathbf{x}}(n)$, $\hat{\mathbf{y}}(n+p)$ and $\mathbf{V}(n+p)$ are available, $\sigma_i(n+p)$ can be estimated as follows:

$$\sigma_i(n+p) = \arg\max_{\sigma_i} \log f\left(\widehat{\mathbf{B}}(n+p)\middle|\mathbf{Q}\right)\Big|_{\mathbf{V}=\mathbf{V}(n+p)}, \tag{35}$$

where $i = 1, 2$. The details of the algorithm for estimating $\sigma_i(n+p)$ are given in [58,59]. Thus, after $\sigma_i(n+p)$ is estimated, $\mathbf{V}(n+p+1)$ can be estimated using Eq. (31).

3.2. Recursive algorithm for updating forecasting model based on numerical output

In the above subsection, we studied the recursive algorithm for online updating the forecasting model under judgmental output. In this subsection, we will develop a new recursive algorithm under numerical output.

In this case, $\hat{\mathbf{y}}$ is a numerical value and mainly refers to the observation of a system obtained by a sensor in this paper. Similarly, we also assume that $\hat{\mathbf{y}}$ is a random variable and $\hat{\mathbf{y}}(1+p),\ldots,\hat{\mathbf{y}}(n+p)$ are independent, so that

$$f(\hat{\mathbf{y}}(1+p),\ldots,\hat{\mathbf{y}}(n+p)|\mathbf{Q}) = \prod_{\tau=1}^{n} f(\hat{\mathbf{y}}(\tau+p)|\mathbf{Q}), \tag{36}$$

where $f(\hat{\mathbf{y}}(\tau+p)|\mathbf{Q})$ is the pdf of $\hat{\mathbf{y}}(\tau+p)$ at time instant $(\tau+p)$.

It is desirable that for a given input, $\hat{\mathbf{x}}(n)$, Eq. (18) can be used to predict an output, $\mathbf{y}(n+p)$, which should be as close to $\hat{\mathbf{y}}(n+p)$ as possible. Here $\mathbf{y}(n+p)$ is considered as the expectation of $\hat{\mathbf{y}}(n+p)$. We assume that $\hat{\mathbf{y}}(n+p)$ obeys the following normal distribution:

$$f(\hat{\mathbf{y}}(n+p)|\mathbf{Q}) = \frac{1}{\sqrt{2\pi\sigma}}\exp\left\{-\frac{(\hat{\mathbf{y}}(n+p) - \mathbf{y}(n+p))^2}{2\sigma}\right\}, \tag{37}$$

where $\mathbf{Q} = [\mathbf{V}^T, \sigma]^T$. σ denotes the variance. $\mathbf{y}(n+p)$ is the average score of the distributed assessment as represented in Eq. (26) at time instant $(n+p)$ and there is [44,45,49]

$$\mathbf{y}(n+p) = \sum_{j=1}^{N} \mu_j \beta_j(n+p), \tag{38}$$

where μ_j denotes the utility (or score) of an individual consequent D_j, either given or estimated by experts. $\beta_j(n+p)$ is also generated by Eq. (26) for a given input $\hat{\mathbf{x}}(n)$.

In order to simplify the calculation, we also assume that the elements of $\mathbf{V}$ are independent of σ. Similar to the first case under judgmental output, two steps are used to estimate $\mathbf{V}$ and σ, respectively.

Step 1: Similar to Eqs. (28)–(31), when we consider only parameter vector $\mathbf{V}$, the following recursive algorithm is obtained

$$\mathbf{V}(n+p+1) = \mathbf{V}(n+p) + \frac{1}{n}\left[\Xi_2'(\mathbf{Q}(n+p))\right]^{-1}\Gamma_2'(\mathbf{Q}(n+p)), \tag{39}$$

where $\mathbf{Q}$ consists of the belief degrees, the rule weights, the attribute weights and σ of the two BRBs. $\Gamma_2'(\mathbf{Q}(n+p))$ and $\Xi_2'(\mathbf{Q}(n+p))$ are the derivatives with respect to $\mathbf{V}$. There are

$$\Gamma_2'(\mathbf{Q}(n+p)) \triangleq \nabla_{\mathbf{V}} \log f(\hat{\mathbf{y}}(n+p)|\mathbf{Q}(n+p)), \tag{40}$$

$$\Xi_2'(\mathbf{Q}(n+p)) \triangleq E\left\{-\nabla_{\mathbf{V}}\nabla_{\mathbf{V}}^T \log f(\hat{\mathbf{y}}(n+p)|\mathbf{Q})|\mathbf{Q}(n+p)\right\}. \tag{41}$$

According to Eq. (37), Eqs. (40) and (41) can be written as follows:

$$\left[\Gamma_2'(\mathbf{Q}(n+p))\right]_a = \frac{(\hat{\mathbf{y}}(n+p) - \mathbf{y}(n+p))}{\sigma(n+p)} \sum_{j=1}^{N} \mu_j \frac{\partial \beta_j(n+p)}{\partial V_a}\bigg|_{\mathbf{V}=\mathbf{V}(n+p)}, \tag{42}$$

$$\left[\Xi_2'(\mathbf{Q}(n+p))\right]_{a,b} = \frac{1}{\sigma(n+p)}\left[\sum_{j=1}^{N} \mu_j \frac{\partial \beta_j(n+p)}{\partial W_a}\right]\left[\sum_{j=1}^{N} \mu_j \frac{\partial \beta_j(n+p)}{\partial W_b}\right]\Bigg|_{\mathbf{V}=\mathbf{V}(n+p)}, \tag{43}$$

where $a = L + M + L \times N + N + N \times N$ and $b = L + M + L \times N + N + N \times N$. The derivatives used in Eqs. (42) and (43) are also given in Appendix B.

Step 2: In Eqs. (42) and (43), $\sigma(n+p)$ is required. If $\hat{\mathbf{x}}(n)$, $\hat{\mathbf{y}}(n+p)$ and $\mathbf{V}(n+p)$ are available, it is estimated by

$$\sigma(n+p) = \arg\max_{\sigma} \log f(\hat{\mathbf{y}}(n+p)|\mathbf{Q})|_{\mathbf{V}=\mathbf{V}(n+p)} = (\hat{\mathbf{y}}(n+p) - \mathbf{y}(n+p))^2\Big|_{\mathbf{V}=\mathbf{V}(n+p)}. \tag{44}$$

Thus, after $\sigma(n+p)$ is estimated, it can be put into Eq. (39) to estimate $\mathbf{V}(n+p+1)$.

3.3. Parameter constraints and expert intervention for the BRB

In BRB_1 and BRB_2, the belief degrees, the rule weights and the attribute weights should satisfy the following equality and inequality constraints. These constraints can be divided into two types: general constraints that any valid BRB must obey, and domain specific constraints that are provided by experts which reflect the relationships among belief rules.

Type 1 constraints: general constraints that any valid BRB must satisfy.

When a BRB is constructed, trained or updated, its parameters must satisfy certain general constraints [45,46]. For example, BRB_1 needs to satisfy the following constraints:

(1) The rule weights are normalized, so that they are between zero and one, i.e.:

$$0 \leqslant \theta_k^1 \leqslant 1, \quad k = 1, \ldots, L. \tag{45}$$

(2) The attribute weights are normalized, so that they are between zero and one, i.e.:

$$0 \leqslant \bar{\delta}_m^1 \leqslant 1, \quad m = 1, \ldots, M. \tag{46}$$

(3) The belief degree can be regarded as subjective probability, and it must not be less than zero or more than one, i.e.:

$$0 \leqslant \beta_{j,k}^1 \leqslant 1, \quad j = 1, \ldots, N, \ k = 1, \ldots, L. \tag{47}$$

(4) If the kth belief rule is complete, its total belief degree in the consequent will be equal to one, otherwise the total belief degree is less than one, i.e.:

$$\sum_{j=1}^{N} \beta_{j,k}^1 \leqslant 1, \quad k = 1, \ldots, L. \tag{48}$$

Type 2 constraints: special constraints that BRB should satisfy.

A system is normally governed by certain forces or physical principles, and its behavior follows certain patterns. Given the trend of inputs to a specific system, experienced domain experts may be able to predict the trend of the outputs of the system [42]. Such knowledge can be extracted and represented in terms of equality and inequality constraints on the consequent belief degrees of rules in both BRB_1 and BRB_2. In BRB_1, for example, the equality constraints $\beta^1_{j,k} = 0$ can be used to represent the knowledge that a consequent D_j in the kth rule will not occur for sure. The inequality constraint $\beta^1_{j,k} \geqslant \beta^1_{j,k+1}$ represents the knowledge that the consequent D_j in the kth rule is more likely to occur than that in the $(k+1)$th rule. Therefore, when such constraints are added into the process in which BRB is manually tuned, trained or updated, we say that the process is under expert intervention. Let $\mathbf{V} = [V_1, \ldots, V_{U+N+N\times N}]^T$ and $U = L + M + L \times N$. Without loss of generality, we assume that the equality and inequality constraints can be presented as follows:

(1) For BRB_1, there are

$$h^1_i(\mathbf{V}) = h^1_i(V_{L+M+1}, \ldots, V_U) = 0, \quad i = 1, \ldots, I, \tag{49}$$
$$z^1_{1,s_1}(\mathbf{V}) = z^1_{1,s_1}(V_{L+M+1}, \ldots, V_U) \leqslant 0, \quad s_1 = 1, \ldots, S_1. \tag{50}$$

(2) For BRB_2, there are

$$h^2_g(\mathbf{V}) = h^2_g(V_{U+N+1}, \ldots, V_{U+N+N\times N}) = 0, \quad g = 1, \ldots, G, \tag{51}$$
$$z^2_{1,w_1}(\mathbf{V}) = z^2_{1,w_1}(V_{U+N+1}, \ldots, V_{U+N+N\times N}) \leqslant 0, \quad w_1 = 1, \ldots, W_1, \tag{52}$$

where I, S_1, G and W_1 denote the numbers of constraints added by experts.

3.4. Recursive algorithms under parameter constraints and expert intervention

In this subsection, the parameter constraints as given in Eqs. (45)–(52) will be taken into account in the recursive algorithms as shown in Eqs. (31) and (39). Firstly, the constraints in Eqs. (45)–(48) are re-written as follows:

(1) For BRB_1, there are

$$z^1_{2,s_2}(\mathbf{V}) = z^1_{2,s_2}\left(V_{L+M+(k-1)\times N+1}, \ldots, V_{L+M+(k-1)\times N+N}\right) \leqslant 0, \quad s_2 = 1, \ldots, S_2, \tag{53}$$
$$z^1_{3,s_3}(\mathbf{V}) = z^1_{3,s_3}(V_{s_3}) = -V_{s_3} \leqslant 0, \quad s_3 = 1, \ldots, S_3, \tag{54}$$
$$z^1_{4,s_4}(\mathbf{V}) = z^1_{4,s_4}(V_{s_4}) = V_{s_4} - 1 \leqslant 0, \quad s_4 = 1, \ldots, S_4, \tag{55}$$

where $z^1_{2,s_2}\left(V_{L+M+(k-1)\times N+1}, \ldots, V_{L+M+(k-1)\times N+N}\right) = \sum_{j=1}^N V_{L+M+(s_2-1)\times N+j} - 1$, $S_2 = L$, $S_3 = S_4 = U$.

(2) For BRB_2, there are

$$z^2_{2,w_2}(\mathbf{V}) = z^2_{2,w_2}\left(V_{U+N+(k-1)\times N+1}, \ldots, V_{U+N+(k-1)\times N+N}\right) \leqslant 0, \quad w_2 = 1, \ldots, W_2, \tag{56}$$
$$z^2_{3,w_3}(\mathbf{V}) = z^2_{3,w_3}(V_{U+w_3}) = -V_{U+w_3} \leqslant 0, \quad w_3 = 1, \ldots, W_3, \tag{57}$$
$$z^2_{4,w_4}(\mathbf{V}) = z^2_{4,w_4}(V_{U+w_4}) = V_{U+w_4} - 1 \leqslant 0, \quad w_4 = 1, \ldots, W_4, \tag{58}$$

where $z^2_{2,w_2}\left(V_{U+N+(k-1)\times N+1}, \ldots, V_{U+N+(k-1)\times N+N}\right) = \sum_{j=1}^N V_{U+N+(w_2-1)\times N+j} - 1$, $W_2 = N$, $W_3 = W_4 = N + N \times N$.

Define

$$\mathbf{h}(\mathbf{V}) \triangleq \left[h^1_1(\mathbf{V}), \ldots, h^1_I(\mathbf{V}), h^2_1(\mathbf{V}), \ldots, h^2_G(\mathbf{V}), Z^1_1(\mathbf{V}), \ldots, Z^1_4(\mathbf{V}), Z^2_1(\mathbf{V}), \ldots, Z^2_4(\mathbf{V})\right]^T, \tag{59}$$

where $Z^i_j(\mathbf{V}) = \left[z^i_{j,1}(\mathbf{V}), \ldots, z^i_{j,S_j}(\mathbf{V})\right]$, $i = 1, 2$ and $j = 1, \ldots, 4$.

Assume

$$\Omega(\mathbf{V}) \triangleq \left\| \left[h^1_1(\mathbf{V}), \ldots, h^1_I(\mathbf{V}), h^2_1(\mathbf{V}), \ldots, h^2_G(\mathbf{V})\right]^T \right\|, \tag{60}$$

$$\tilde{z}^1_{j,s_j}(\mathbf{V}) \triangleq \max\left[0, z^1_{j,s_j}(\mathbf{V})\right] \quad \text{and} \quad \Psi^1_j(\mathbf{V}) = \sum_{s_j=1}^{S_j} \left[\tilde{z}^1_{j,s_j}(\mathbf{V})\right]^2, \tag{61}$$

$$\tilde{z}^2_{j,w_j}(\mathbf{V}) \triangleq \max\left[0, z^2_{j,w_j}(\mathbf{V})\right] \quad \text{and} \quad \Psi^2_j(\mathbf{V}) = \sum_{w_j=1}^{W_j} \left[\tilde{z}^2_{j,w_j}(\mathbf{V})\right]^2, \tag{62}$$

where the values of I, G, S_j and W_j are the same as Eqs. (49)–(58) and $j = 1, \ldots, 4$. $\|\cdot\|$ denotes Euclidean norm. $\Omega(\mathbf{V})$ and $\Psi^i_j(\mathbf{V})$ $(i = 1, 2;\ j = 1, \ldots, 4)$ are penalty functions and are used to deal with the constraints of two BRBs.

Define

$$\boldsymbol{\rho}(\mathbf{V}) \triangleq \left[\frac{\partial \Omega(\mathbf{V})}{\partial V_1}, \ldots, \frac{\partial \Omega(\mathbf{V})}{\partial V_{U+N+N\times N}}\right]^T, \tag{63}$$

$$\boldsymbol{\phi}_j^i(\mathbf{V}) \triangleq \left[\frac{\partial \Psi_j^i(\mathbf{V})}{\partial V_1}, \ldots, \frac{\partial \Psi_j^i(\mathbf{V})}{\partial V_{U+N+N\times N}}\right]^T, \tag{64}$$

where i = 1, 2 and j = 1,...,4.

Assume that $\mathbf{I}_u$ is the identity matrix whose dimension is u. According to [19], the recursive algorithms as given in Eqs. (31) and (39) can be revised as the following form to deal with the constraints in Eqs. (45)–(52)

$$\mathbf{V}(n+p+1) = \mathbf{V}(n+p) + \frac{\alpha}{n}\left\{\boldsymbol{\pi}(\mathbf{V}(n+p))\left[\Xi'_k(\mathbf{Q}(n+p)) + \gamma\mathbf{I}_{U+N+N\times N}\right]^{-1}\Gamma'_k(\mathbf{Q}(n+p)) - \frac{K}{2}\boldsymbol{\phi}(\mathbf{V}(n+p))\right\}, \tag{65}$$

where $\boldsymbol{\phi}(\mathbf{V}(n+p)) = \boldsymbol{\rho}(\mathbf{V}(n+p)) + \sum_{i=1}^{2}\sum_{j=1}^{4}\boldsymbol{\phi}_j^i(\mathbf{V}(n+p))$. The step factor α can change the convergence speed and there is $\alpha \geqslant 1$. Since not all of the rules in the two BRBs will be activated and the matrix $\Xi'_k(\mathbf{Q}(n+p))$ $(k = 1,2)$ may be singular at time instant $(n+p), \Xi'_k(\mathbf{Q}(n+p))$ is amended using $\gamma\mathbf{I}_{U+N+N\times N}$ with $\gamma \geqslant 0$ so that it becomes positive definite. K denotes a positive real number and its value may change from case to case. Here we choose $K = (n + p)$ [19]. $\boldsymbol{\pi}(\mathbf{V}(n + p))$ denotes the value of $\boldsymbol{\pi}(\mathbf{V})$ at time instant $(n + p)$, and there is

$$\boldsymbol{\pi}(\mathbf{V}) = \mathbf{I}_{U+N\times N+N} - \mathbf{H}(\mathbf{V})^T(\mathbf{H}(\mathbf{V})\mathbf{H}(\mathbf{V})^T)^{-1}\mathbf{H}(\mathbf{V}), \tag{66}$$

where $\boldsymbol{\pi}$ and $\mathbf{H}(\mathbf{V})$ denote the projection operator and the Jacobian matrix of $\mathbf{h}(\mathbf{V})$, respectively.

The recursive algorithms in Eq. (65) can also be written as the following general form

$$\mathbf{V}(n+p+1) = \prod_H \left\{\mathbf{V}(n+p) + \frac{\alpha}{n}\left[\Xi'_k(\mathbf{Q}(n+p))\right]^{-1}\Gamma'_k(\mathbf{Q}(n+p))\right\}, \quad k = 1,2, \tag{67}$$

where H is a constraint set composed of the constraints as given in Eqs. (45)–(52), and $\prod_H\{\cdot\}$ is the projection onto H. Thus, the elements of $\mathbf{V}(n + p + 1)$ as shown in Eq. (67) satisfy the constraints in Eqs. (45)–(52).

As a result of the above discussion, under expert intervention, the procedure of the recursive algorithms for updating the parameters of the forecasting model can be summarized as follows:

Step 1: According to domain specific physical principles and historical information, experts can estimate patterns about system dynamic behavior. Such patterns are converted into constraints as shown in Eqs. (49)–(52).

Step 2: Let n = 0. Assign initial values to the parameter vector $\mathbf{V}(n + p)$ and the covariance $\boldsymbol{\chi}(n + p)$ or the variance $\sigma(n + p)$. Moreover, $\mathbf{V}(n + p)$ contains all the parameters of BRB_1 and BRB_2 which satisfy the constraints in Eqs. (45)–(48). Here p is the forecasting step.

Step 3: Given the input $\hat{\mathbf{x}}(n)$ and the output $\hat{\mathbf{y}}(n + p)$ of a system, the recursive algorithms in Eq. (67) are used to estimate $\mathbf{V}(n + 1 + p)$ which satisfies the constraints in Eqs. (45)–(52). Then the covariance matrix $\boldsymbol{\chi}(n + 1 + p)$ or the variance $\sigma(n + 1 + p)$ is estimated using Eqs. (35) or (44) which are required by Eqs. (33) or (43).

Step 4: After the input $\hat{\mathbf{x}}(n+1)$ and output $\hat{\mathbf{y}}(n+1+p)$ are available, let $n = n + 1$ and go to Step 3. Otherwise, go to Step 5.

Step 5: Once the forecasting model, composed of BRB_1 and BRB_2, is updated, the system behavior after p steps can be predicted according to the given input $\hat{\mathbf{x}}(n)$ at time instant n.

Remark 1. From Eqs. (32), (33), (42) and (43), we see that the analytical formulations of $\Xi'_k(\mathbf{Q}(n+p))$ and $\Gamma'_k(\mathbf{Q}(n+p))$ can be obtained. On the other hand, the appropriate initial parameters of the forecasting model may be obtained after they are manually tuned by experts, which is useful to improve the convergence speed of the recursive algorithms. This is important when there is high real-time requirement for updating the forecasting model.

Remark 2. Under the equality and inequality constraints, Eq. (67) is indeed a stochastic approximation algorithm whose convergence was proven [20]. We can also prove the convergence of the algorithm as given in Eq. (67) if the appropriate initial values of the parameters are given. The convergence of the algorithm will be discussed later in the simulation study of Section 4.1.4.

4. A numerical simulation study and a case study

4.1. A numerical simulation study

4.1.1. Problem formulation

Here a system with two tanks is used to show the effectiveness of the proposed forecasting model and recursive algorithms. This system is shown in Fig. 1.

In Fig. 1, water flows into the first tank at the rate Q_1 (m^3/s), then to the second tank at the rate Q_{12} (m^3/s) and finally out of the second tank at the rate Q_{20} (m^3/s). Suppose that a slow jam fault happens at the outlet of tank 2. The dynamic model of the two tanks is given as follows:

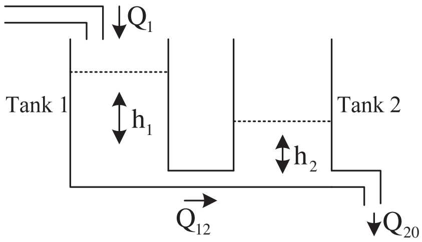

Fig. 1. The layout of two tanks.

$$\begin{cases} A\dot{h}_1 = Q_1 - Q_{12}, \\ A\dot{h}_2 = Q_{12} - Q_{20}, \end{cases} \tag{68}$$

with

$$\begin{cases} Q_{12} = a_1 s\, sgn(h_1 - h_2)\sqrt{2g|h_1 - h_2|}, \\ Q_{20} = a_2 s\sqrt{2gh_2}, \end{cases} \tag{69}$$

where a_i and h_i ($i = 1,2$) are the outflow coefficients and the liquid level in meters (m) of the two tanks, and $sgn(z)$ is the sign of the argument z. s is the section area of the connection pipe (m^2). A is the section area of the two tanks (m^2) which are of the same size. The technical parameters of the two tanks are given in Table 1. T_s is the sampling interval.

We assume that a_2 decreases with time as follows:

$$\phi(n) = a_2 - 0.001n, \tag{70}$$

where $\phi(n)$ is the outflow coefficient of the second tank when the system is in the slow jam fault. Eq. (70) shows that the fault is getting worse.

In order to verify the proposed forecasting model and recursive algorithms, we use the levels of the two tanks shown in Fig. 2 as the inputs of the forecasting model, and the outflow coefficient φ as the output. In other words, the two levels are considered as the antecedent attributes and the outflow coefficient is the consequent in BRB_1.

Table 1
The technical parameters of the two tanks.

$A = 0.15\ m^2$	$s = 0.00005\ m^2$	$Q_1 = 0.00005\ m^3/s$	$T_s = 1$ s
$g = 9.8\ cm/s^2$	$a_1 = 0.4$	$a_2 = 0.5$	

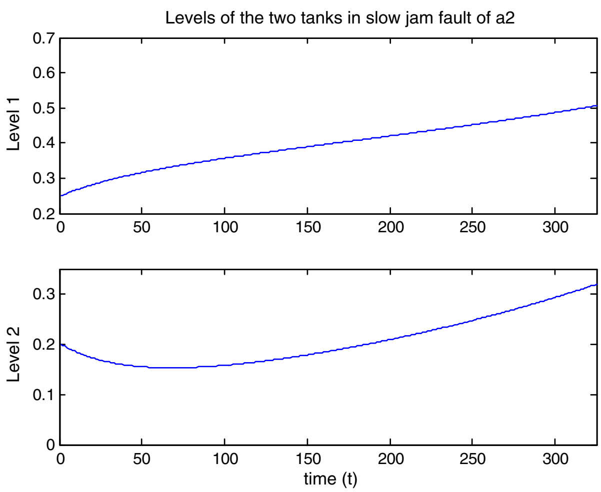

Fig. 2. The levels of the two tanks during the slow jam fault.

4.1.2. Rules

Let *Level 1* and *Level 2* denote the two levels, OC^1 the outflow coefficient at the current time instant and OC^2 the outflow coefficient after p steps. In order to predict the fault of the two tanks, a forecasting model including BRB_1 and BRB_2 needs to be constructed. BRB_1 is used to capture the relationship between *Level 1* and *Level 2* through OC^1 at the current time instant, which captures the dynamics of the two tanks. BRB_2 captures the relationship between OC^1 and OC^2, which can be considered as the forecasting model and used to predict the future behavior of the system. Thus, the belief rules of BRB_1 and BRB_2 can be represented as follows:

$$\begin{aligned}&R_k : \text{If } Level\ 1(n) \text{ is } A_1^k \wedge Level\ 2(n) \text{ is } A_2^k, \quad \text{then } OC^1(n) \text{ is } \left\{(D_1, \beta_{1,k}^1), (D_2, \beta_{2,k}^1)\right\}\\ &\text{with a rule weight } \theta_k^1 \text{ and attribute weight } \delta_{1,k}^1, \delta_{2,k}^1, \ \left(\sum_{i=1}^{2} \beta_{i,k}^1 \leqslant 1\right)\end{aligned} \tag{71}$$

$$\begin{aligned}&R_l' : \text{If } OC^1(n) \text{ is } D_l, \quad \text{then } OC^2(n+p) \text{ is } \left\{(D_1, \beta_{1,l}^2), (D_2, \beta_{2,l}^2)\right\}\\ &\text{with a rule weight } \theta_l^2 \text{ and attribute weight } \delta_{1,l}^2, \ \left(\sum_{i=1}^{2} \beta_{i,l}^2 \leqslant 1\right)\end{aligned} \tag{72}$$

where R_k ($k = 1,\ldots,7$) and R_l' ($l = 1,2$) are the belief rules of BRB_1 and BRB_2, respectively. In BRB_1, A_1^k and A_2^k ($k = 1,\ldots,7$) are the referential points of *Level 1* and *Level 2*, respectively. In BRB_2, D_l ($l = 1,2$) are the referential points of OC^1. n and p denote the current time instant and the forecasting step, respectively. As mentioned in Section 2.3, there is always $\delta_{1,l}^2 = 1$. In the following discussion, we assume $\delta_j^1 = \delta_{j,k}^1$ ($j = 1,2$).

4.1.3. Referential points of the antecedents and the consequent

In BRB_1, three referential points are used for the antecedents *Level 1* and *Level 2*, and they are low (L), medium (M), and high (H). That is $A_i^k \in \{L,M,H\}$, where $i = 1, 2$. For the consequent OC^1, two referential points are assumed and they are normal (N) and fault (F). That is $\mathbf{D} = (D_1,D_2) = (N,F)$. In BRB_2, the referential points for the consequent $OC^2$are assumed to be the same as for OC^1.

The referential points defined above are in linguistic terms and need to be quantified. The quantified results are listed in Tables 2–4, respectively.

4.1.4. Simulation results based on judgmental output

In order to verify the recursive algorithm under judgmental output, the following simulation is carried out.

In this simulation, three models are constructed for the validation analysis. From the two inputs *Level 1* and *Level 2* and certain parameters of the two BRBs, the first one is constructed and used as a benchmark model to generate data which is used to examine and test the proposed behavior prediction method. The second one is the forecasting model, which is given initially by an expert and then updated using the proposed algorithm and the data generated by the benchmark model, leading to the third optimally trained forecasting model.

Step 1: Construct a benchmark model and generate its judgmental output.

Suppose that there is a real system to be modeled by a benchmark model which is composed of two BRBs. The values of θ_k^1, δ_j^1 and θ_l^2 are all assumed to be 1, where $k = 1,\ldots,7$, $j = 1,2$, $i = 1,2$ and $l = 1,2$. Tables 5 and 6 represent BRB_1 and BRB_2, respectively. Here the forecasting step is chosen as $p = 10$. The belief degrees $\hat{\beta}_{j,k}^1$ and $\hat{\beta}_{i,l}^2$ can be determined as follows, which are listed in Tables 5 and 6 for the current time instant and 10 steps afterwards, respectively.

For example, if *Level 1* is H (high or 0.55) and *Level 2* is H (high or 0.32) at time $n = 325$ s, then the value of the outflow coefficient at time $n = 325$ s can be represented as a distributed assessment $OC = \{(N,0),(F,1)\}$ using the information transformation technique as given in Appendix A. Therefore, a belief rule can be set as follows:

$$\text{If } Level\ 1 \text{ is } H \quad \text{and} \quad Level\ 2 \text{ is } H, \quad \text{then } OC \text{ is } \{(N,0),(F,1)\}, \tag{73}$$

which is shown in the last row of Table 5. In other words, we have $\hat{\beta}_{1,7} = 0$ and $\hat{\beta}_{2,7} = 1$.

Similarly, the other belief rules in Tables 5 and 6 can be generated. Since the combinations "L and H" and "H and L" may not happen according to prior knowledge, these combinations are not given in Table 5.

Table 2
The referential points of *Level 1*.

Linguistic terms	L	M	H
Numerical values (m)	0.2	0.5	0.55

Table 3
The referential points of *Level 2*.

Linguistic terms	*L*	*M*	*H*
Numerical values (m)	0	0.25	0.32

Table 4
The referential points of OC^1.

Linguistic terms	*N*	*F*
Numerical values (m)	0.5	0.175

Table 5
Belief rules of BRB_1 in the benchmark model.

Rule number	*Level 1* and *Level 2*	OC^1 distribution $\{D_1, D_2\} = \{N, F\}$
1	*L* and *L*	$\{(D_1, 0.99), (D_2, 0.01)\}$
2	*L* and *M*	$\{(D_1, 0.7), (D_2, 0.3)\}$
3	*M* and *L*	$\{(D_1, 0.55), (D_2, 0.45)\}$
4	*M* and *M*	$\{(D_1, 0.5), (D_2, 0.5)\}$
5	*M* and *H*	$\{(D_1, 0.4), (D_2, 0.6)\}$
6	*H* and *M*	$\{(D_1, 0.2), (D_2, 0.8)\}$
7	*H* and *H*	$\{(D_1, 0), (D_2, 1)\}$

To apply the belief rules of the two BRBs in the benchmark model, the input values *Level 1*(n) and *Level 2*(n), which denote the water levels in tanks 1 and 2 at time instant n, need to be transformed and represented in terms of the referential values defined in Tables 2 and 3. Similar process is also required in the traditional rule based system modeling approach. The detailed transformation processes are given in Appendix A.

After the transformation of the input, Eq. (14) is used to calculate the judgmental output of the benchmark model according to the parameters given in Tables 5 and 6. The judgmental output is composed of the belief degrees to the normal and fault states of the two tanks.

In this simulation, 325 datasets are generated, including the input values *Level 1* and *Level 2*, and the judgmental output of the benchmark model. Moreover, the anterior 315 and last 10 datasets are chosen as the training and testing datasets, respectively. In order to use the proposed recursive algorithm, it is assumed that the judgmental output is a random variable with a normal distribution.

Step 2: Determine the special behavioral constraints that the two BRBs should satisfy.

According to the referential values of the antecedent attributes in BRB_1 given in Table 5 and the patterns that the two tanks behave, the belief degree to the linguistic term "fault (F)" should increase when the rule number increases as the seriousness of the fault increases. In BRB_2 as given in Table 6, from belief rule 1 to belief rule 2, the belief degree to "F" should also increase. Thus, the following constraints should be satisfied by the parameters of the trained model

$$\beta^1_{1,J_1} \geqslant \beta^1_{1,J_2}, \quad J_1 = 1, \ldots, 6,\ J_2 = J_1 + 1, \tag{74}$$
$$\beta^1_{2,J_1} \leqslant \beta^1_{2,J_2}, \quad J_1 = 1, \ldots, 6,\ J_2 = J_1 + 1, \tag{75}$$
$$\beta^1_{1,J_3} \geqslant \beta^1_{2,J_3}, \quad J_3 = 1, \ldots, 3, \tag{76}$$
$$\beta^1_{1,J_4} \leqslant \beta^1_{2,J_4}, \quad J_4 = 4, \ldots, 7, \tag{77}$$
$$\beta^2_{1,1} \geqslant \beta^2_{1,2}, \quad \beta^2_{2,1} \leqslant \beta^2_{2,2}, \tag{78}$$
$$\beta^2_{1,1} \geqslant \beta^2_{2,1}, \quad \beta^2_{1,2} \leqslant \beta^2_{2,2}, \tag{79}$$

where Eqs. (74), (75) and (78) represent the inequality relationships among the rules in the BRB system that we are going to establish. Eqs. (76), (77) and (79) represent the inequality relationships among the consequent belief degrees in a rule. Eqs. (74)–(77) are used for BRB_1, and Eqs. (78) and (79) for BRB_2. For example, in Eq. (74), the belief degree $\beta^1_{1,1}$ to "normal (N)"in the first rule of BRB_1 is larger than $\beta^1_{1,2}$, the belief degree to the same referential point in the second rule, which shows that rule 1 represents a more "N" state of the system than rule 2 does. In Eq. (79), the belief degree $\beta^2_{1,2}$ to "N" is smaller than $\beta^2_{2,2}$, the belief degree to "F", which shows that in the second rule of BRB_2, the possibility of the system being "N" is smaller than being "F". In a similar way, we can explain the meanings of the other constraints.

Table 6
Belief rules of BRB_2 in the benchmark model.

Rule number	OC^1	OC^2 distribution $\{D_1, D_2\} = \{N, F\}$
1	D_1	$\{(D_1, 0.8), (D_2, 0.2)\}$
2	D_2	$\{(D_1, 0.01), (D_2, 0.99)\}$

Step 3: Set initial parameters of the forecasting model.

The initial belief degrees of BRB_1 and BRB_2, which are listed in Tables 7 and 8, respectively, are given by an expert according to the running patterns and historical data of the two tanks. For example, if *Level 1* is *L* and *Level 2* is *M*, the expert judges that the possibility of system is in the normal state is larger than the fault state according to the historical information. Moreover, the expert assesses that the belief degree to "*N*" is 0.74 and the belief degree to "*F*" is 0.26. Then a belief rule is obtained as the third row of Table 7. In addition, the initial values of θ_k^1, δ_j^1 and θ_l^2 are all set to 1, where $k = 1,\ldots,7$, $j = 1,2$ and $l = 1,2$. In this way, the initial BRB_1 and BRB_2 are constructed as an initial forecasting model.

From Fig. 3, we see that the judgmental output generated by the initial forecasting model change between 0 and 1, which shows that the forecasting model is capable of capturing continuous causal relationship between antecedents and consequents. However, in the traditional IF-THEN rule, the consequent is either 100% true or 100% false. Such a rule base has limited capacity in representing knowledge in a real world.

As shown in Fig. 3, however, we observe that the judgmental output generated by Eq. (14) and the initial forecasting model do not match the training data generated by the benchmark model. This means that the initial rule base provided by the expert is approximately right, but not good enough. Therefore, it is necessary to use the available information to fine tune and update the rule base.

Step 4: Update the forecasting model initially constructed in Step 3.

Using the training datasets which include the input values *Level 1*(n) and *Level 2*(n), and the judgmental output at time instant $(n + p)$ generated by the benchmark model in Step 1, the constraints as given in Step 2 and the initial belief rules in Step 3, the proposed recursive algorithm under judgmental output shown in Eq. (67) is applied to update the initial forecasting model, where $n = 1,\ldots,305$. The updated BRB_1 and BRB_2 including their rule weights are listed in Tables 9 and 10, respectively. As shown in Fig. 3, the forecasting judgmental output generated by the updated model can match the training data more closely than the initial model established by experts without online tuning. In addition, as shown in Tables 9 and 10, the two updated BRB systems can satisfy the general constraints in Eqs. (45)–(48) and the domain specific behavioral constraints in Eqs. (74)–(79).

Step 5: Test the updated forecasting model.

In order to examine the forecasting ability of the updated model, the testing datasets are used. Figs. 4 and 5 give the testing data and the forecasting judgmental output at time instant $(n + p)$ for the antecedent values *Level 1*(n) and *Level 2*(n), where $n = 306,\ldots,315$. Fig. 6 gives the absolute error (AE) [22] between the testing data and the forecasting judgmental output generated by the initial model and the AE between the testing data and the forecasting judgmental output generated by the updated model.

Table 7
Initial belief rules of BRB_1 provided by an expert.

Rule number	*Level 1* and *Level 2*	OC^1 distribution $\{D_1, D_2\} = \{N, F\}$
1	*L* and *L*	$\{(D_1, 0.94), (D_2, 0.06)\}$
2	*L* and *M*	$\{(D_1, 0.74), (D_2, 0.26)\}$
3	*M* and *L*	$\{(D_1, 0.51), (D_2, 0.49)\}$
4	*M* and *M*	$\{(D_1, 0.5), (D_2, 0.5)\}$
5	*M* and *H*	$\{(D_1, 0.43), (D_2, 0.57)\}$
6	*H* and *M*	$\{(D_1, 0.24), (D_2, 0.76)\}$
7	*H* and *H*	$\{(D_1, 0), (D_2, 1)\}$

Table 8
Initial belief rules of BRB_2 provided by an expert.

Rule number	OC^1	OC^2 distribution $\{D_1, D_2\} = \{N, F\}$
1	D_1	$\{(D_1, 0.8), (D_2, 0.2)\}$
2	D_2	$\{(D_1, 0.1), (D_2, 0.9)\}$

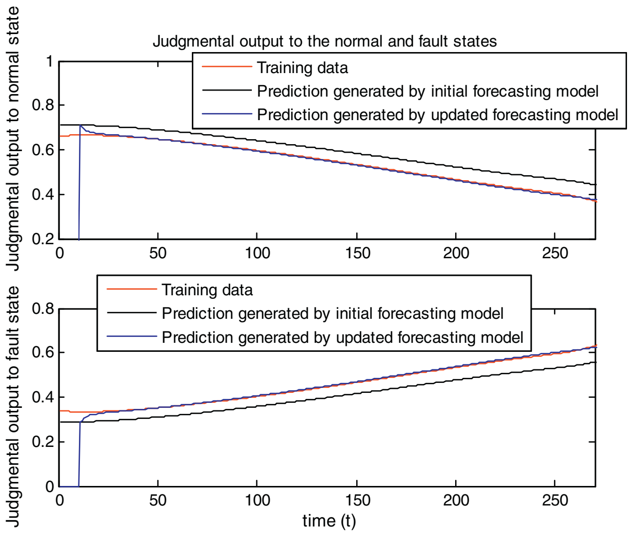

Fig. 3. Training data generated by the benchmark model and the estimated judgmental outputs by the initial forecasting model given by an expert and the updated forecasting model.

Table 9
Updated belief rules and rule weights of BRB_1 based on judgmental output.

Rule number	Updated rule weight	*Level 1* and *Level 2*	OC^1 distribution $\{D_1, D_2\} = \{N, F\}$
1	0.9571	*L* and *L*	$\{(D_1, 0.9086), (D_2, 0.0914)\}$
2	0.9990	*L* and *M*	$\{(D_1, 0.7142), (D_2, 0.2858)\}$
3	0.9989	*M* and *L*	$\{(D_1, 0.5548), (D_2, 0.4452)\}$
4	0.9910	*M* and *M*	$\{(D_1, 0.4898), (D_2, 0.5102)\}$
5	1.0000	*M* and *H*	$\{(D_1, 0.4257), (D_2, 0.5743)\}$
6	1.0000	*H* and *M*	$\{(D_1, 0.2524), (D_2, 0.7476)\}$
7	1.0000	*H* and *H*	$\{(D_1, 0.0308), (D_2, 0.9692)\}$

Table 10
Updated belief rules and rule weights of BRB_2 based on judgmental output.

Rule number	Updated rule weight	OC^1	OC^2 distribution $\{D_1, D_2\} = \{N, F\}$
1	0.7640	D_1	$\{(D_1, 0.8367), (D_2, 0.1633)\}$
2	0.9916	D_2	$\{(D_1, 0.0670), (D_2, 0.9330)\}$

These three figures show that (1) compared with the initial model, the forecasting values of the updated model can match the testing data more closely. (2) The AE between the testing data and the forecasting values generated by the initial model is larger than that between the testing data and the forecasting values generated by the updated model. It is obvious that the updated model can predict the future behavior more accurately than the initial one.

Step 6: Convergence analysis.

It has been pointed out that if the appropriate initial values of parameters are chosen, the proposed recursive algorithm can converge to a locally optimal point [20]. In order to study the performance of the algorithm, the mean squared error (MSE) [22] is chosen to measure the estimation accuracy. MSE may be represented as:

$$\mathrm{MSE} \triangleq \sum_{k=1}^{L}\sum_{j=1}^{N}\left(\beta_{j,k}^{1}(n) - \hat{\beta}_{j,k}^{1}\right)^2 + \sum_{l=1}^{N}\sum_{s=1}^{N}\left(\beta_{s,l}^{2}(n) - \hat{\beta}_{s,l}^{2}\right)^2, \tag{80}$$

where $\hat{\beta}_{j,k}^{1}$ and $\hat{\beta}_{s,l}^{2}$ are the belief degrees of BRB_1 and BRB_2 in the benchmark model, respectively. $\beta_{j,k}^{1}(n)$ and $\beta_{s,l}^{2}(n)$ are the estimates generated by the recursive algorithm at time instant n.

As shown in Fig. 7, if the appropriate initial parameters which are close to the values of the benchmark model are chosen, the estimations generated by the proposed recursive algorithm converge to the parameters of the benchmark model very quickly.

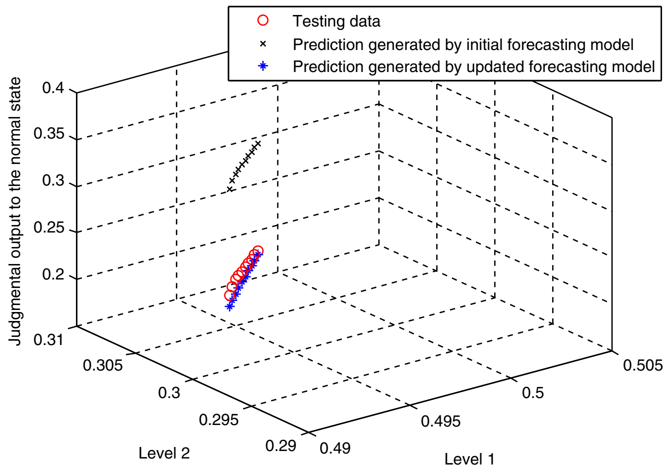

Fig. 4. Testing data and the forecasting judgmental outputs to the normal state generated by the initial and updated models.

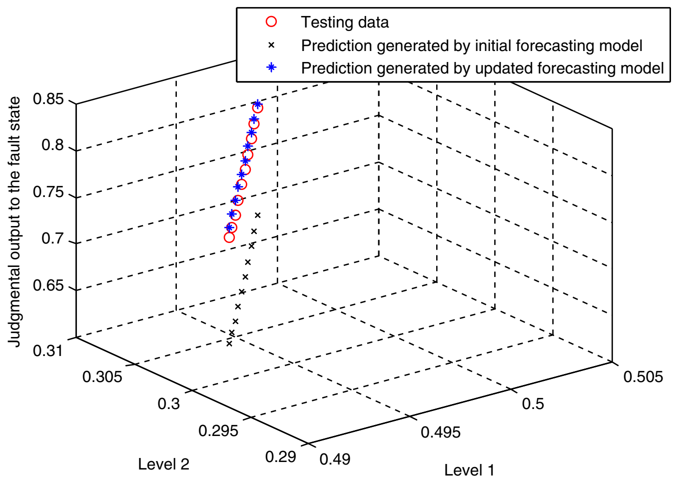

Fig. 5. Testing data and the forecasting judgmental outputs to the fault state generated by the initial and updated models.

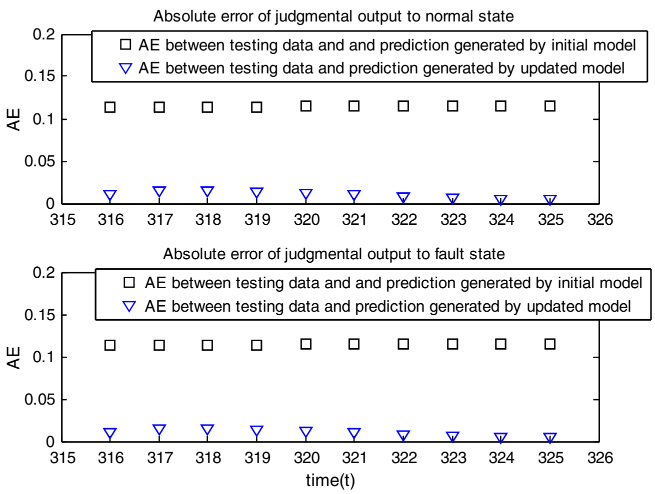

Fig. 6. AEs between testing data and predictions generated by the initial and updated models.

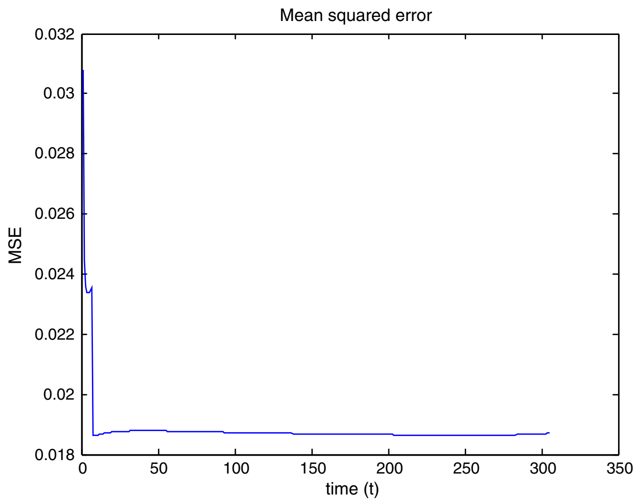

Fig. 7. MSE between the belief degrees in the benchmark model and the estimated values.

4.1.5. Simulation results based on numerical output

In the above subsection, the proposed recursive algorithm under judgmental output has been verified. In this subsection, the following simulation is used to verify the proposed recursive algorithm under numerical output. Different from the first simulation, the outflow coefficient, $\phi(n)$, generated by Eq. (70) is chosen as the real output of the forecasting model.

In this simulation, 325 datasets are used, including the input values *Level 1*(n) and *Level 2*(n) generated by Eqs. (68) and (69) and the output value $\phi(n)$, where $n = 1, \ldots, 325$. Similarly, the anterior 315 and last 10 datasets are chosen as the training and testing datasets, respectively. In order to use the proposed recursive algorithm, it is assumed that ϕ is a random variable with the normal distribution.

Step 1: Set initial parameters.

The initial belief degrees, rule weights and attribute weights are the same as those in the case of judgmental output. We also choose the forecasting step as $p = 10$. As shown in Fig. 8, the forecasting values after p steps given by the initial model, OC^2, do not match the real values, i.e., the training data. This means that the initial forecasting model, composed of the two BRBs which are provided by the expert, is indeed rather inaccurate. Therefore, it is necessary to update the initial forecasting model.

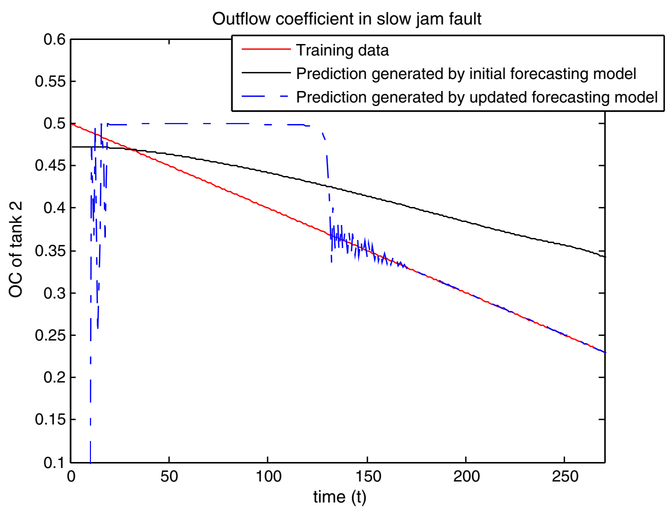

Fig. 8. Training data and the estimated OC^2 by the initial and updated forecasting models in slow jam fault.

Step 2: Update the initial forecasting model.

When the forecasting model is updated, the two levels *Level 1*(n) and *Level 2*(n), and the outflow coefficient $\phi(n+p)$ are used as the input and output values in the training data, where $n = 1,\ldots,305$.

After *Level 1*(n) and *Level 2*(n) are transformed and represented in terms of the referential values using the transformation technique as given in Appendix A, the recursive algorithm under numerical output in Eq. (67) and the constraints in Eqs. (74)–(79) is used to update the expert defined forecasting model. The updated belief degrees and rule weights of the two BRBs are given in Tables 11 and 12. We can see that the updated belief degrees satisfy the general constraints in Eqs. (45)–(48) and the special behavioral constraints in Eqs. (74)–(79). Figs. 8 and 9 show that the updated forecasting model can predict the future behavior more accurately than the initial model after the algorithm converged at about time 150.

Step 3: Test the updated forecasting model.

Similarly, the testing datasets are used to validate the forecasting ability. Fig. 10 gives the testing data at time instant $(n+p)$ and prediction, i.e., the estimated OC^2 at time instant $(n+p)$ for the antecedent values *Level 1*(n) and *Level 2*(n), where $n = 306,\ldots,315$. Fig. 11 gives the AE between the testing data and predictions generated by the initial and updated forecasting models. It is also shown that compared with the initial model, the updated one can predict the outflow coefficient more precisely.

4.1.6. Concluding remarks

From the above numerical study, it can be concluded that the initial forecasting model given by the experts are not accurate and the proposed recursive algorithms can be used to update it whether the output of the BRB is judgmental or numerical. The updated forecasting model can predict the future behavior accurately. Moreover, if an expert can provide judgments about the running patterns of a system, the updated forecasting model can also satisfy those judgments, which are referred to as expert intervention. It is illustrated using examples that if the appropriate initial values of the parameters are given, the recursive algorithms can converge fairly fast.

4.2. A case study

In order to demonstrate the potential application of the proposed forecasting model in engineering, we apply the scheme to build a model for forecasting leak size from an oil pipeline using data taken from an operational long distance oil pipeline installed in Great Britain.

4.2.1. Problem formulation

When a leak develops in a pipeline, flow and pressure will change following certain patterns in the pipeline. Similar to [42], the difference between inlet flow and outlet flow, the average pipeline pressure change over time and the leak rate, denoted by *FlowDiff*, *PressureDiff* and *LeakSize*, respectively, are used as the leak datasets.

In the leak trial, 322 sample leak datasets were collected at the rate of 10 s per sample from no leak to 25% leak. Note that 25% leak denotes that if there is 100 kg oil in the pipeline, the leak will be 25 kg oil. When the leak changes from zero to 25%, Figs. 12 and 13 give the curves for *FlowDiff* and *PressureDiff*, respectively.

Table 11
Updated belief rules and rule weights of BRB_1 based on numerical output.

Rule number	Updated rule weight	*Level 1* and *Level 2*	OC^1 distribution $\{D_1, D_2\} = \{N, F\}$
1	0.8913	L and L	$\{(D_1, 0.8546), (D_2, 0.1454)\}$
2	0.7934	L and M	$\{(D_1, 0.8545), (D_2, 0.1455)\}$
3	0.9997	M and L	$\{(D_1, 0.6615), (D_2, 0.3385)\}$
4	0.9564	M and M	$\{(D_1, 0.4592), (D_2, 0.5408)\}$
5	1.0000	M and H	$\{(D_1, 0.3712), (D_2, 0.6288)\}$
6	1.0000	H and M	$\{(D_1, 0.1254), (D_2, 0.8746)\}$
7	1.0000	H and H	$\{(D_1, 0.1250), (D_2, 0.8750)\}$

Table 12
Updated belief rules and rule weights of BRB_2 based on numerical output.

Rule number	Updated rule weight	OC^1	OC^2 distribution $\{D_1, D_2\} = \{N, F\}$
1	0.1809	D_1	$\{(D_1, 0.9990), (D_2, 0.0010)\}$
2	1.0000	D_2	$\{(D_1, 0.0010), (D_2, 0.9990)\}$

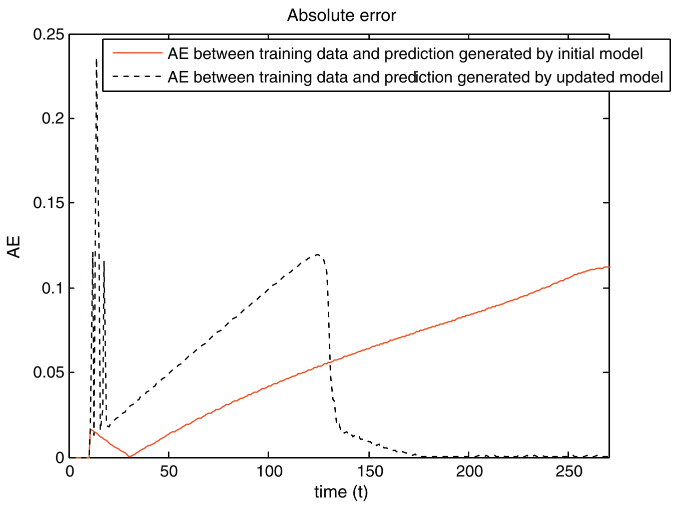

Fig. 9. AEs between the training data and the estimated values generated by the initial and updated forecasting model in slow jam fault.

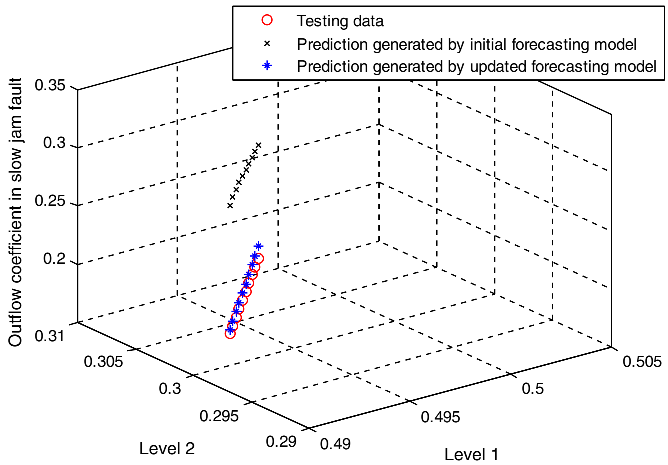

Fig. 10. Testing data and predictions generated by the initial and updated forecasting models.

In order to verify the proposed forecasting model and recursive algorithms, based on the collected leak data and the initial forecasting model given by experts, one part of the leak datasets will be used to update the initial forecasting model, and another part to test the updated forecasting model.

4.2.2. Construction of the initial forecasting model

Since *FlowDiff* and *PressureDiff* are the two very important factors in detecting whether there is leak in the pipeline, they are chosen as the antecedent attributes of the rule base. Obviously, *LeakSize* is the consequent of the BRB.

Let *LeakSize* (n) denote the leak rate at the current time instant n and *LeakSize* $(n + p)$ the leak rate after p steps. In order to predict the leak rate of the pipeline, a forecasting model composed of BRB_1 and BRB_2 needs to be constructed. BRB_1 is used to capture the relationship between *FlowDiff* (n) and *PressureDiff* (n) through *LeakSize* (n) at current time instant n, which captures the dynamics of the oil pipeline. BRB_2 captures the relationship between *LeakSize* (n) and *LeakSize* $(n + p)$, which can be considered as the forecasting model and used to predict the future behavior of the system.

In order to construct two BRBs, the antecedents and consequent should be given some referential points. Similar to [42], we choose these points as follows:

For *FlowDiff*, eight referential points are used and they are negative large (NL), negative medium (NM), negative small (NS), negative very small (NVS), zero (Z), positive small (PS), positive medium (PM), and positive large (PL), i.e.:

$$A_1^k \in \{\mathrm{NL}, \mathrm{NM}, \mathrm{NS}, \mathrm{NVS}, Z, \mathrm{PS}, \mathrm{PM}, \mathrm{PL}\}. \tag{81}$$

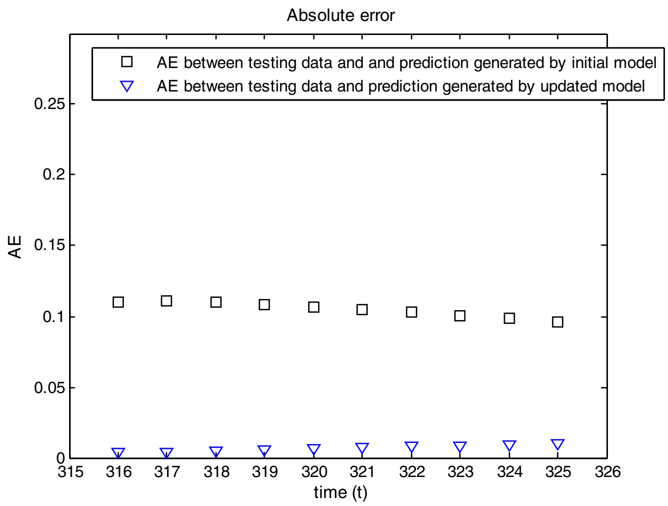

Fig. 11. AEs between testing data and predictions generated by the initial and updated forecasting models.

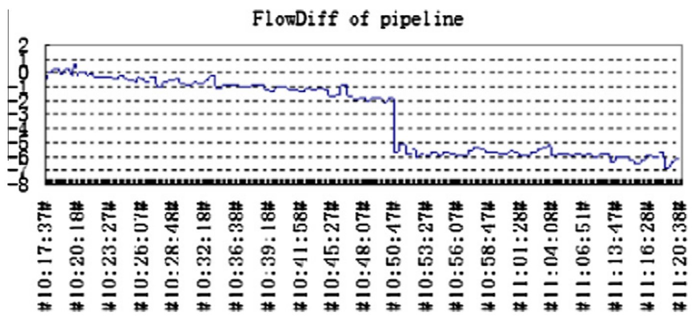

Fig. 12. The *FlowDiff* of the pipeline.

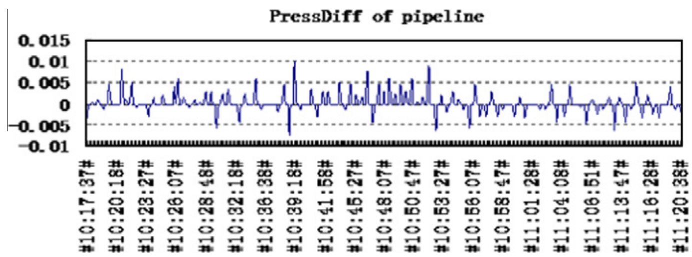

Fig. 13. The *PressureDiff* of the pipeline.

For *PressureDiff*, seven referential points are used and they are NL, NM, NS, *Z*, PS, PM, PL, i.e.:

$$A_2^k \in \{\mathrm{NL}, \mathrm{NM}, \mathrm{NS}, Z, \mathrm{PS}, \mathrm{PM}, \mathrm{PL}\}. \quad (82)$$

For *LeakSize*, five referential points are used: zero (*Z*), very small (VS), medium (*M*), high (*H*) and very high (VH), i.e.:

$$\mathbf{D} = (D_1, D_2, D_3, D_4, D_5) = (Z, \mathrm{VS}, M, H, \mathrm{VH}). \quad (83)$$

The referential points defined above for the antecedent attributes and consequent are in linguistic terms and need to be quantified. The quantified results are given in Tables 13–15, respectively.

Thus, according to the operational patterns of an oil pipeline, an expert can construct BRB_1 and BRB_2. Belief rules in BRB_1 and BRB_2 are represented as follows:

$$\begin{aligned} &R_k : \text{If } \mathit{FlowDiff}(n) \text{ is } A_1^k \wedge \mathit{PressureDiff}(n) \text{ is } A_2^k \\ &\quad \text{Then } \mathit{LeakSize}(n) \text{ is } \left\{ \left(D_1, \beta_{1,k}^1\right), \left(D_2, \beta_{2,k}^1\right), \left(D_3, \beta_{3,k}^1\right), \left(D_4, \beta_{4,k}^1\right), \left(D_5, \beta_{5,k}^1\right) \right\} \\ &\quad \text{With a rule weight } \theta_k^1 \text{ and attribute weight } \delta_{1,k}^1, \delta_{2,k}^1, \left(\sum_{i=1}^{5} \beta_{i,k} \leqslant 1 \right) \end{aligned} \quad (84)$$

Table 13
The referential points of *FlowDiff*.

Linguistic terms	NL	NM	NS	NVS	Z	PS	PM	PL
Numerical values	−10	−5	−3	−1	0	1	2	3

Table 14
The referential points of *PressureDiff*.

Linguistic terms	NL	NM	NS	Z	PS	PM	PL
Numerical values	−0.042	−0.025	−0.01	0	0.01	0.025	0.042

Table 15
The referential points of *LeakSize*.

Linguistic terms	Z	VS	M	H	VH
Numerical values	0	2	4	6	8

$$\begin{aligned}&R'_l: \text{If } \mathit{LeakSize}(n) \text{ is } D_l,\\&\quad \text{Then } \mathit{LeakSize}(n+p) \text{ is } \left\{\left(D_1,\beta^2_{1,l}\right),\left(D_2,\beta^2_{2,l}\right),\left(D_3,\beta^2_{3,l}\right),\left(D_4,\beta^2_{4,l}\right),\left(D_5,\beta^2_{5,l}\right)\right\}\\&\quad \text{With a rule weight } \theta^2_l \text{ and attribute weight } \delta^2_{1,l}, \left(\sum_{i=1}^{5}\beta^2_{i,l}\leqslant 1\right)\end{aligned}\tag{85}$$

where R_k (k = 1,...,56) and R'_l ($l = 1,\ldots,5$) are the belief rules of BRB_1 and BRB_2, respectively. In BRB_1, A_1^k and A_2^k are the referential points of *FlowDiff* and *PressureDiff* as given in Eqs. (81) and (82), respectively. In BRB_2, D_l (l = 1,...,5) are the referential points of *LeakSize*. n and p denote the current time instant and the forecasting step, respectively. As mentioned in Section 2.3, there is always $\delta^2_{1,l} = 1$. In the following discussions, suppose $\delta^1_j = \delta^1_{j,k}$ $(j = 1,2)$.

The values of belief degrees, rule weights and attribute weights can be provided by an expert to construct the initial BRB_1 and BRB_2. Here the initial belief degrees of BRB_1 are the same as shown in Table A1 of Ref. [58]. For convenience of the following discussions, the initial BRB_1 is provided in Table 16. The initial belief degrees of BRB_2 are given in Table 17. Moreover, the parameters θ^1_k, θ^2_l and δ^1_j are all assumed to be 1, for k = 1,...,56, l = 1,...,5 and j = 1,2. The initial forecasting model is composed of the initial BRB_1 and BRB_2.

4.2.3. Specific behavioral constraints on pipeline leak given by human experts

(1) Specific behavioral constraints that BRB_ 1 should satisfy.
According to the sampled data, mass balance principle and the operational patterns of a pipeline, it can be deduced that for the first seven rules of BRB_1, as given in Table 16, leak size should get smaller. In other words, the belief degrees to the linguistic term D_1 should get larger, where D_1 means that there is no leak. If BRB_ 1 is divided into eight groups and each group includes seven rules, there are similar relationships among the seven rules in each group. These patterns can be captured by the following inequalities [58]

$$\beta^1_{i,1} \leqslant \beta^1_{i,2} \leqslant \cdots \leqslant \beta^1_{i,7}, \quad i = 1,\ldots,8. \tag{86}$$

Moreover, the linguistic term D_5 denotes that leak is the most severe. According to the sampled data and expert knowledge, D_5 in some rules can not occur, which can be described as follows [58]:

$$\beta^1_{j,5} = 0, \quad j = 5,\ldots,56. \tag{87}$$

Apart from Eqs. (86) and (87), other equalities and inequalities can also be added if necessary, depending upon whether there is other expert knowledge available.

(2) Specific behavioral constraints that BRB_ 2 should satisfy.
Similarly, from BRB_2 as given in Table 17, the following constraints can be obtained:

$$\beta^2_{1,1} \geqslant \beta^2_{1,2}, \quad \beta^2_{5,4} \leqslant \beta^2_{5,5}, \tag{88}$$

$$\beta_{1,k} = 0, \quad k = 3,4,5, \tag{89}$$

$$\beta_{5,k} = 0, \quad k = 1,2,3. \tag{90}$$

Table 16
Initial belief rules of BRB 1 provided by an expert.

Rule number	*FlowDiff* and *PressureDiff*	*LeakSize* distribution $\{D_1,D_2,D_3,D_4,D_5\} = \{0,2,4,6,8\}$
1	NL and NL	$\{(D_1,0),(D_2,0),(D_3,0),(D_4,0),(D_5,1)\}$
2	NL and NM	$\{(D_1,0),(D_2,0),(D_3,0),(D_4,0.3),(D_5,0.7)\}$
3	NL and NS	$\{(D_1,0),(D_2,0),(D_3,0.2),(D_4,0.8),(D_5,0)\}$
4	NL and *Z*	$\{(D_1,0),(D_2,0),(D_3,0.8),(D_4,0.2),(D_5,0)\}$
5	NL and PS	$\{(D_1,0.65),(D_2,0.35),(D_3,0),(D_4,0),(D_5,0)\}$
6	NL and PM	$\{(D_1,0.85),(D_2,0.15),(D_3,0),(D_4,0),(D_5,0)\}$
7	NL and PL	$\{(D_1,0.95),(D_2,0.05),(D_3,0),(D_4,0),(D_5,0)\}$
8	NM and NL	$\{(D_1,0),(D_2,0),(D_3,0.1),(D_4,0.9),(D_5,0)\}$
9	NM and NM	$\{(D_1,0),(D_2,0),(D_3,0.7),(D_4,0.3),(D_5,0)\}$
10	NM and NS	$\{(D_1,0),(D_2,0.7),(D_3,0.3),(D_4,0),(D_5,0)\}$
11	NM and *Z*	$\{(D_1,0),(D_2,0.9),(D_3,0.1),(D_4,0),(D_5,0)\}$
12	NM and PS	$\{(D_1,0.8),(D_2,0.2),(D_3,0),(D_4,0),(D_5,0)\}$
13	NM and PM	$\{(D_1,0.9),(D_2,0.1),(D_3,0),(D_4,0),(D_5,0)\}$
14	NM and PL	$\{(D_1,0.99),(D_2,0.01),(D_3 0),(D_4,0),(D_5,0)\}$
15	NS and NL	$\{(D_1,0),(D_2,0),(D_3,0.4),(D_4,0.6),(D_5,0)\}$
16	NS and NM	$\{(D_1,0),(D_2,0),(D_3,0.8),(D_4,0.2),(D_5,0)\}$
17	NS and NS	$\{(D_1,0),(D_2,0.3),(D_3,0.6),(D_4,0.1),(D_5,0)\}$
18	NS and *Z*	$\{(D_1,0.1),(D_2,0.7),(D_3,0.2),(D_4,0),(D_5,0)\}$
19	NS and PS	$\{(D_1,0.7),(D_2,0.3),(D_3,0),(D_4,0),(D_5,1)\}$
20	NS and PM	$\{(D_1,0.9),(D_2,0.1),(D_3,0),(D_4,0),(D_5,1)\}$
21	NS and PL	$\{(D_1,1),(D_2,0),(D_3,0),(D_4,0),(D_5,0)\}$
22	NVS and NL	$\{(D_1,0.02),(D_2,0.11),(D_3,0.39),(D_4,0.48),(D_5,0)\}$
23	NVS and NM	$\{(D_1,0.1),(D_2,0.78),(D_3,0.12),(D_4,0),(D_5,0)\}$
24	NVS and NS	$\{(D_1,0.36),(D_2,0.64),(D_3,0),(D_4,0),(D_5,0)\}$
25	NVS and *Z*	$\{(D_1,1),(D_2,0),(D_3,0),(D_4,0),(D_5,0)\}$
26	NVS and PS	$\{(D_1,1),(D_2,0),(D_3,0),(D_4,0),(D_5,0)\}$
27	NVS and PM	$\{(D_1,1),(D_2,0),(D_3,0),(D_4,0),(D_5,0)\}$
28	NVS and PL	$\{(D_1,1),(D_2,0),(D_3,0),(D_4,0),(D_5,0)\}$
29	*Z* and NL	$\{(D_1,1),(D_2,0),(D_3,0),(D_4,0),(D_5,0)\}$
30	*Z* and NM	$\{(D_1,1),(D_2,0),(D_3,0),(D_4,0),(D_5,0)\}$
31	*Z* and NS	$\{(D_1,1),(D_2,0),(D_3,0),(D_4,0),(D_5,0)\}$
32	*Z* and *Z*	$\{(D_1,1),(D_2,0),(D_3,0),(D_4,0),(D_5,0)\}$
33	*Z* and PS	$\{(D_1,1),(D_2,0),(D_3,0),(D_4,0),(D_5,0)\}$
34	*Z* and PM	$\{(D_1,1),(D_2,0),(D_3,0),(D_4,0),(D_5,0)\}$
35	*Z* and PL	$\{(D_1,1),(D_2,0),(D_3,0),(D_4,0),(D_5,0)\}$
36	PS and NL	$\{(D_1,0.39),(D_2,0.61),(D_3,0),(D_4,0),(D_5,0)\}$
37	PS and NM	$\{(D_1,0.9),(D_2,0.1),(D_3,0),(D_4,0),(D_5,0)\}$
38	PS and NS	$\{(D_1,1),(D_2,0),(D_3,0),(D_4,0),(D_5,0)\}$
39	PS and *Z*	$\{(D_1,1),(D_2,0),(D_3,0),(D_4,0),(D_5,0)\}$
40	PS and PS	$\{(D_1,1),(D_2,0),(D_3,0),(D_4,0),(D_5,0)\}$
41	PS and PM	$\{(D_1,1),(D_2,0),(D_3,0),(D_4,0),(D_5,0)\}$
42	PS and PL	$\{(D_1,1),(D_2,0),(D_3,0),(D_4,0),(D_5,0)\}$
43	PM and NL	$\{(D_1,0.1),(D_2,0.9),(D_3,0),(D_4,0),(D_5,0)\}$
44	PM and NM	$\{(D_1,0.3),(D_2,0.7),(D_3,0),(D_4,0),(D_5,0)\}$
45	PM and NS	$\{(D_1,0.85),(D_2,0.15),(D_3,0),(D_4,0),(D_5,0)\}$
46	PM and *Z*	$\{(D_1,0.98),(D_2,0.02),(D_3,0),(D_4,0),(D_5,0)\}$
47	PM and PS	$\{(D_1,1),(D_2,0),(D_3,0),(D_4,0),(D_5,0)\}$
48	PM and PM	$\{(D_1,1),(D_2,0),(D_3,0),(D_4,0),(D_5,0)\}$
49	PM and PL	$\{(D_1,1),(D_2,0),(D_3,0),(D_4,0),(D_5,0)\}$
50	PL and NL	$\{(D_1,0.9),(D_2,0.1),(D_3,0),(D_4,0),(D_5,0)\}$
51	PL and NM	$\{(D_1,0.99),(D_2,0.01),(D_3,0),(D_4,0),(D_5,0)\}$
52	PL and NS	$\{(D_1,1),(D_2,0),(D_3,0),(D_4,0),(D_5,0)\}$
53	PL and *Z*	$\{(D_1,1),(D_2,0),(D_3,0),(D_4,0),(D_5,0)\}$
54	PL and PS	$\{(D_1,1),(D_2,0),(D_3,0),(D_4,0),(D_5,0)\}$
55	PL and PM	$\{(D_1,1),(D_2,0),(D_3,0),(D_4,0),(D_5,0)\}$
56	PL and PL	$\{(D_1,1),(D_2,0),(D_3,0),(D_4,0),(D_5,0)\}$

4.2.4. Online updating the initial forecasting model and comparative study

In order to update the initial forecasting model as constructed in Section 4.2.2, the first 305 leak datasets are used as the training data. During the training process, *FlowDiff (n)* and *PressureDiff (n)* are used as the input values, *and LeakSize (n + p)* as the output value, where $n = 1,\ldots,300$. According to the practical need, the forecasting step is chosen as $p = 5$.

After the input values *FlowDiff (n)* and *PressureDiff (n)* are transformed and represented in terms of the referential values as defined in Tables 13 and 14 at time instant n using the technique as given in Appendix A, the recursive algorithm under numerical output as given in Eq. (67) is used to update the initial forecasting model under the constraints given in Eqs. (45)–(48) and (86)–(90). The updated belief degrees and rule weights of the forecasting model are given in Tables 18 and 19. It can

Table 17
Initial belief rules of BRB_2 provided by an expert.

Rule number	*LeakSize*	*LeakSize_P* distribution $\{D_1, D_2, D_3, D_4, D_5\} = \{0, 2, 4, 6, 8\}$
1	D_1	$\{(D_1, 0.5), (D_2, 0.5), (D_3, 0), (D_4, 0), (D_5, 0)\}$
2	D_2	$\{(D_1, 0.05), (D_2, 0.35), (D_3, 0.6), (D_4, 0), (D_5, 0)\}$
3	D_3	$\{(D_1, 0), (D_2, 0.1), (D_3, 0.4), (D_4, 0.5), (D_5, 0)\}$
4	D_4	$\{(D_1, 0), (D_2, 0), (D_3, 0.05), (D_4, 0.35), (D_5, 0.6)\}$
5	D_5	$\{(D_1, 0), (D_2, 0), (D_3, 0), (D_4, 0.05), (D_5, 0.95)\}$

Table 18
Updated belief rules and rule weights of BRB_1 under special behavioral constraints.

Rule number	Updated rule weight	*FlowDiff* and *PressureDiff*	*LeakSize* distribution $\{D_1, D_2, D_3, D_4, D_5\} = \{0, 2, 4, 6, 8\}$
1	1	NL and NL	$\{(D_1, 0.0009), (D_2, 0), (D_3, 0), (D_4, 0), (D_5, 0.9991)\}$
2	1	NL and NM	$\{(D_1, 0.0748), (D_2, 0), (D_3, 0), (D_4, 0.2739), (D_5, 0.6513)\}$
3	1	NL and NS	$\{(D_1, 0.3818), (D_2, 0.0001), (D_3, 0.1230), (D_4, 0.4845), (D_5, 0.0096)\}$
4	0.975	NL and *Z*	$\{(D_1, 0.4652), (D_2, 0.0010), (D_3, 0.2850), (D_4, 0.2020), (D_5, 0.0467)\}$
5	0.9974	NL and PS	$\{(D_1, 0.6432), (D_2, 0.3501), (D_3, 0.0003), (D_4, 0.0063), (D_5, 0)\}$
6	1	NL and PM	$\{(D_1, 0.8498), (D_2, 0.1502), (D_3, 0), (D_4, 0), (D_5, 0)\}$
7	1	NL and PL	$\{(D_1, 0.9500), (D_2, 0.0500), (D_3, 0), (D_4, 0), (D_5, 0)\}$
8	1	NM and NL	$\{(D_1, 0.0009), (D_2, 0), (D_3, 0.0999), (D_4, 0.8992), (D_5, 0)\}$
9	1	NM and NM	$\{(D_1, 0.0790), (D_2, 0), (D_3, 0..6374), (D_4, 0.2836), (D_5, 0)\}$
10	0.973	NM and NS	$\{(D_1, 0..3792), (D_2, 0.4322), (D_3, 0.1308), (D_4, 0.0577), (D_5, 0)\}$
11	1	NM and *Z*	$\{(D_1, 0.3882), (D_2, 0.0112), (D_3, 0.0015), (D_4, 0.5992), (D_5, 0)\}$
12	0.9867	NM and PS	$\{(D_1, 0.7775), (D_2, 0.1942), (D_3, 0.0001), (D_4, 0.0284), (D_5, 0)\}$
13	1	NM and PM	$\{(D_1, 0.8999), (D_2, 0.1001), (D_3, 0), (D_4, 0), (D_5, 0)\}$
14	1	NM and PL	$\{(D_1, 0.9900), (D_2, 0.0100), (D_3, 0), (D_4, 0), (D_5, 0)\}$
15	1	NS and NL	$\{(D_1, 0.0009), (D_2, 0), (D_3, 0.3995), (D_4, 0.5996), (D_5, 0)\}$
16	1	NS and NM	$\{(D_1, 0.0240), (D_2, 0), (D_3, 0.7773), (D_4, 0.1987), (D_5, 0)\}$
17	1	NS and NS	$\{(D_1, 0.0393), (D_2, 0.2837), (D_3, 0.5775), (D_4, 0.0995), (D_5, 0)\}$
18	0.9726	NS and *Z*	$\{(D_1, 0.0775), (D_2, 0.5483), (D_3, 0.2349), (D_4, 0.1392), (D_5, 0)\}$
19	0.9983	NS and PS	$\{(D_1, 0.6899), (D_2, 0.2957), (D_3, 0.0045), (D_4, 0.0100), (D_5, 0)\}$
20	1	NS and PM	$\{(D_1, 0.8999), (D_2, 0.1001), (D_3, 0), (D_4, 0), (D_5, 0)\}$
21	1	NS and PL	$\{(D_1, 1), (D_2, 0), (D_3, 0), (D_4, 0), (D_5, 0)\}$
22	1	NVS and NL	$\{(D_1, 0.0111), (D_2, 0.1114), (D_3, 0.3943), (D_4, 0.4832), (D_5, 0)\}$
23	1	NVS and NM	$\{(D_1, 0.0920), (D_2, 0.7877), (D_3, 0.1203), (D_4, 0), (D_5, 0)\}$
24	0.9672	NVS and NS	$\{(D_1, 0.3786), (D_2, 0.6152), (D_3, 0.0027), (D_4, 0.0034), (D_5, 0)\}$
25	1	NVS and *Z*	$\{(D_1, 0.6793), (D_2, 0.0007), (D_3, 0.1187), (D_4, 0.2013), (D_5, 0)\}$
26	0.9994	NVS and PS	$\{(D_1, 0.9773), (D_2, 0.0003), (D_3, 0.0086), (D_4, 0.0137), (D_5, 0)\}$
27	1	NVS and PM	$\{(D_1, 1), (D_2, 0), (D_3, 0), (D_4, 0), (D_5, 0)\}$
28	1	NVS and PL	$\{(D_1, 1), (D_2, 0), (D_3, 0), (D_4, 0), (D_5, 0)\}$
29	1	*Z* and NL	$\{(D_1, 0.9969), (D_2, 0.0010), (D_3, 0.0017), (D_4, 0.0004), (D_5, 0)\}$
30	1	*Z* and NM	$\{(D_1, 0.9975), (D_2, 0.0010), (D_3, 0.0006), (D_4, 0.0009), (D_5, 0)\}$
31	1	*Z* and NS	$\{(D_1, 0.9976), (D_2, 0.0010), (D_3, 0.0004), (D_4, 0.0010), (D_5, 0)\}$
32	1	*Z* and *Z*	$\{(D_1, 1), (D_2, 0), (D_3, 0), (D_4, 0), (D_5, 0)\}$
33	1	*Z* and PS	$\{(D_1, 1), (D_2, 0), (D_3, 0), (D_4, 0), (D_5, 0)\}$
34	1	*Z* and PM	$\{(D_1, 1), (D_2, 0), (D_3, 0), (D_4, 0), (D_5, 0)\}$
35	1	*Z* and PL	$\{(D_1, 1), (D_2, 0), (D_3, 0), (D_4, 0), (D_5, 0)\}$
36	1	PS and NL	$\{(D_1, 0.3855), (D_2, 0.6145), (D_3, 0), (D_4, 0), (D_5, 0)\}$
37	1	PS and NM	$\{(D_1, 0.8999), (D_2, 0.1001), (D_3, 0), (D_4, 0), (D_5, 0)\}$
38	1	PS and NS	$\{(D_1, 0.9971), (D_2, 0.0006), (D_3, 0.0011), (D_4, 0.0012), (D_5, 0)\}$
39	1	PS and *Z*	$\{(D_1, 0.9975), (D_2, 0.0010), (D_3, 0.0006), (D_4, 0.0009), (D_5, 0)\}$
40	1	PS and PS	$\{(D_1, 1), (D_2, 0), (D_3, 0), (D_4, 0), (D_5, 0)\}$
41	1	PS and PM	$\{(D_1, 1), (D_2, 0), (D_3, 0), (D_4, 0), (D_5, 0)\}$
42	1	PS and PL	$\{(D_1, 1), (D_2, 0), (D_3, 0), (D_4, 0), (D_5, 0)\}$
43	1	PM and NL	$\{(D_1, 0.0919), (D_2, 0.9081), (D_3, 0), (D_4, 0), (D_5, 0)\}$
44	1	PM and NM	$\{(D_1, 0.2943), (D_2, 0.7056), (D_3, 0), (D_4, 0), (D_5, 0)\}$
45	1	PM and NS	$\{(D_1, 0.8498), (D_2, 0.1502), (D_3, 0), (D_4, 0), (D_5, 0)\}$
46	1	PM and *Z*	$\{(D_1, 0.9800), (D_2, 0.0200), (D_3, 0), (D_4, 0), (D_5, 0)\}$
47	1	PM and PS	$\{(D_1, 1), (D_2, 0), (D_3, 0), (D_4, 0), (D_5, 0)\}$
48	1	PM and PM	$\{(D_1, 1), (D_2, 0), (D_3, 0), (D_4, 0), (D_5, 0)\}$
49	1	PM and PL	$\{(D_1, 1), (D_2, 0), (D_3, 0), (D_4, 0), (D_5, 0)\}$
50	1	PL and NL	$\{(D_1, 0.8999), (D_2, 0.1001), (D_3, 0), (D_4, 0), (D_5, 0)\}$
51	1	PL and NM	$\{(D_1, 0.9900), (D_2, 0.0100), (D_3, 0), (D_4, 0), (D_5, 0)\}$
52	1	PL and NS	$\{(D_1, 1), (D_2, 0), (D_3, 0), (D_4, 0), (D_5, 0)\}$
53	1	PL and *Z*	$\{(D_1, 1), (D_2, 0), (D_3, 0), (D_4, 0), (D_5, 0)\}$
54	1	PL and PS	$\{(D_1, 1), (D_2, 0), (D_3, 0), (D_4, 0), (D_5, 0)\}$
55	1	PL and PM	$\{(D_1, 1), (D_2, 0), (D_3, 0), (D_4, 0), (D_5, 0)\}$
56	1	PL and PL	$\{(D_1, 1), (D_2, 0), (D_3, 0), (D_4, 0), (D_5, 0)\}$

be seen that the updated belief degrees satisfy the general constraints in Eqs. (45)–(48) and the specific behavioral constraints in Eqs. (86)–(90). Moreover, the training is fairly fast using the recursive algorithm.

In order to demonstrate the validity of the proposed algorithm, the following comparative study is carried out.

According to the input values *FlowDiff* (n) and *PressureDiff* (n) and the initial forecasting model constructed in Section 4.2.2, the forecasting values can be generated by Eq. (18), where $n = 1, \ldots, 300$. Fig. 14 displays the training data and the estimated *LeakSize* generated by the initial forecasting model. As shown in Fig. 14, it is obvious that the estimated *LeakSize* generated by the initial forecasting model do not match the training data. This is due to the fact that the expert can not manually determine the accurate parameters in the forecasting model composed of two BRBs, especially for BRB_1. Therefore, it is necessary to update the forecasting model by training, for example using the proposed recursive algorithm.

Fig. 15 shows the training data and the estimated *LeakSize* generated by the updated forecasting model. From Fig. 15, it can be seen that the estimated *LeakSize* generated by the updated model match the training data fairly accurately.

Table 19
Updated belief rules and rule weights of BRB_2 under special behavioral constraints.

Rule number	Updated rule weight	*LeakSize*	*LeakSize_P* distribution $\{D_1, D_2, D_3, D_4, D_5\} = \{0, 2, 4, 6, 8\}$
1	0.7243	D_1	$\{(D_1, 0.1137), (D_2, 0.0827), (D_3, 0.2406), (D_4, 0.5630), (D_5, 0)\}$
2	0.8479	D_2	$\{(D_1, 0.0010), (D_2, 0.1252), (D_3, 0.3327), (D_4, 0.5411), (D_5, 0)\}$
3	0.9930	D_3	$\{(D_1, 0), (D_2, 0.0981), (D_3, 0.3222), (D_4, 0.5797), (D_5, 0)\}$
4	1.0000	D_4	$\{(D_1, 0), (D_2, 0.0855), (D_3, 0.1668), (D_4, 0.3180), (D_5, 0.4297)\}$
5	1.0000	D_5	$\{(D_1, 0), (D_2, 0.0005), (D_3, 0.0013), (D_4, 0.0354), (D_5, 0.9628)\}$

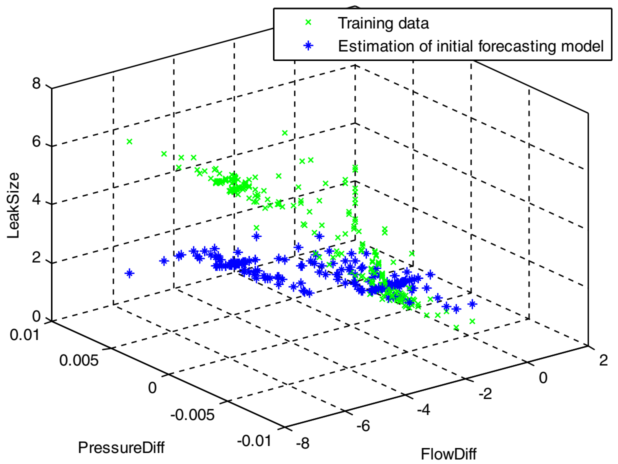

Fig. 14. Training data and estimation generated by initial forecasting model.

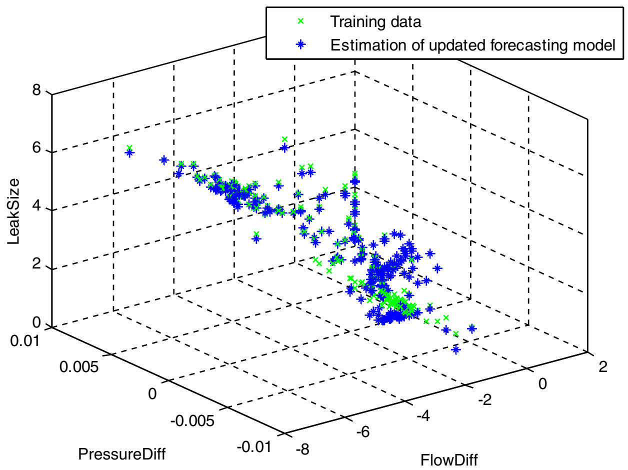

Fig. 15. Training data and estimation generated by updated forecasting model.

Fig. 16 shows the absolute error (AE) between the true leak and the estimated *LeakSize* generated by the initial forecasting model and the AE between the true leak and the estimated *LeakSize* generated by the updated model, respectively. From Fig. 16, it can be seen that the latter is close to zero and smaller than the former.

In order to further demonstrate the proposed recursive algorithm, the mean squared error (MSE) is used [22]. The MSE between the true leak and the estimated *LeakSize* generated by the initial forecasting model is 4.1212. On the other hand, the MSE between the true leak and the estimated *LeakSize* generated by the updated model is 0.3581. The MSE generated by the updated forecasting model is much smaller than the initial one.

From the above comparative study, it can be concluded that the proposed recursive algorithm can update the initial forecasting model online and the updated forecasting model can not only satisfy the general and special constraints, but also predict the training data accurately.

4.2.5. Testing the updated forecasting model and comparative studies

In order to test the forecasting ability of the updated forecasting model, the last 17 leak datasets are used as the testing data and the following comparative study is carried out.

According to the input values *FlowDiff* (n) and *PressureDiff* (n) and the initial and updated forecasting models, the forecasting values can be generated by Eq. (18), where $n = 301,\ldots,317$. Fig. 17 shows the testing data and the forecasting *LeakSize* generated by the initial and updated models for the same antecedent values *FlowDiff* (n) and *PressureDiff* (n), where $n = 301,\ldots,317$. Fig. 18 displays the testing data and the forecasting *LeakSize* on the time scale. In these two Figures, it can be seen that compared with the initial model, the forecasting *LeakSize* generated by the updated model can match the testing data more accurately.

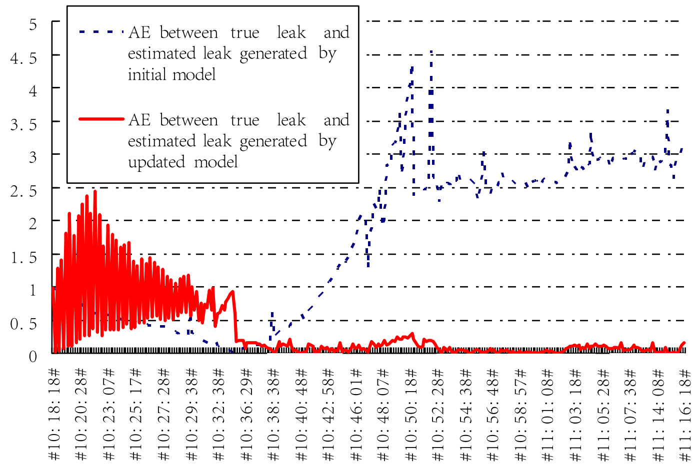

Fig. 16. AEs between true leak and estimations generated by the initial and updated forecasting models.

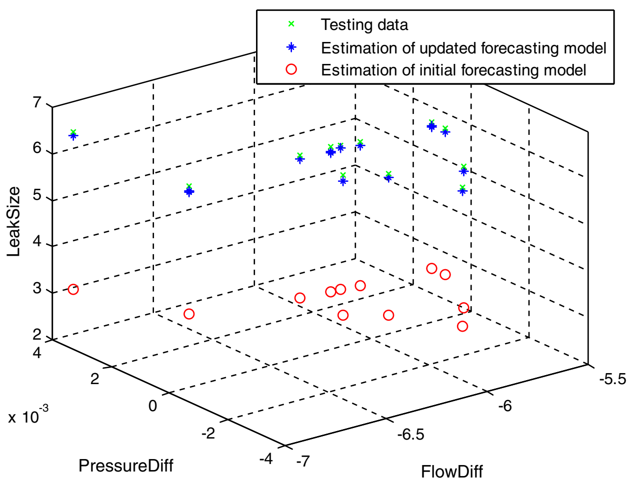

Fig. 17. Testing data and estimations generated by initial and updated forecasting models.

In order to further demonstrate the proposed recursive algorithm, the following comparative studies are constructed.

(1) Mean squared error (MSE).
The MSE between the testing data and the forecasting *LeakSize* generated by the initial model is 9.4103. On the other hand, the MSE between the testing data and the forecasting *LeakSize* generated by the updated model is only 0.01, which is much smaller.
(2) Statistical testing.
In order to show if differences produced by the initial and updated models are statistically meaningful, 17 testing data, 17 forecasting values generated by the initial model and 17 forecasting values generated by the updated model as shown in Fig. 17 are denoted by Sample 1, Sample 2 and Sample 3, respectively. Then the Lilliefors test is used to test if these three small samples obey the normal distributions. Finally the *F*-test is used to test if there is any difference between the standard deviations of two samples. If the standard deviations of two samples are sufficiently similar, the *t*-test is further used to test if there is difference between the means of two samples.

In the following statistical tests, choose the significance level of $\alpha = 0.05$. From the *F*-table and *t*-table, there are

$$F_{\frac{\alpha}{2}}(17-1,17-1) = F_{0.025}(16,16) \approx 2.74, \quad (91)$$

$$t_{\frac{\alpha}{2}}(17+17-2) = t_{0.025}(32) = 2.0369. \quad (92)$$

According to Eq. (91), there is

$$F_{1-\frac{\alpha}{2}}(16,16) = F_{0.975}(16,16) = \frac{1}{F_{0.025}(16,16)} \approx 0.3650. \quad (93)$$

(i) Test if the samples obey the normal distribution.
Using the function lillietest.m of MATLAB that can perform the Lilliefors test, it can be concluded that Sample 1, Sample 2 and Sample 3 all obey the normal distributions. Since the foundation of the *F*-test and the *t*-test is that the sample obeys the normal distribution, then these two tests will be carried out.
(ii) Test the standard deviations of Sample 1 and Sample 2.
Give the null hypothesis: $H_1 : \sigma_1 = \sigma_2$, where σ_1 and σ_2 denote the standard deviations of Sample 1 and Sample 2, respectively. The value of *F*-statistic for Sample 1 and Sample 2 is given by $F_{1,2} = 0.0274$. Obviously, there is $F_{1,2} < F_{0.975}(16,16)$. In other words, the null hypothesis is rejected and there is significant difference between the standard deviations of Sample 1 and Sample 2. Thus, there is no need to carry out the *t*-test.
(iii) Test the standard deviations of Sample 1 and Sample 3.
Give the null hypothesis: $H_2 : \sigma_1 = \sigma_3$, where σ_3 denotes the standard deviation of Sample 3. The value of *F*-statistic for Sample 1 and Sample 3 is given by $F_{1,3} = 0.6785$. Obviously, there is $F_{0.975}(16,16) < F_{1,3} < F_{0.025}(16,16)$. In other words, the null hypothesis is accepted and there is no significant difference between the standard deviations of Sample 1 and Sample 3. Therefore, the *t*-test needs to be carried out.
(iv) Test the means of Sample 1 and Sample 3.
Give the null hypothesis: $H_3 : \mu_1 = \mu_3$, where μ_1 and μ_3 denote the means of Sample 1 and Sample 3, respectively. The value of *t*-statistic for the two samples is given by $T = 1.8974$. Obviously, there is $T < t_{0.025}(32)$. In other words, the null hypothesis is accepted and there is no significant difference between the means of Sample 1 and Sample 3.

From the above comparative analysis, it can be concluded that the forecasting ability of the model updated by the proposed recursive algorithm is much better than the initial model.

4.2.6. Comparative study

In order to demonstrate the validity of the proposed algorithm further, the recursive algorithm under numerical output in Eq. (67) and the first 305 leak datasets are also used to update the initial forecasting model when the specific behavioral

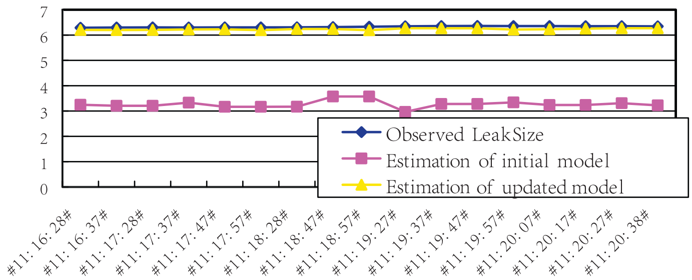

Fig. 18. Testing data and estimations generated by initial and updated forecasting models.

Table 20
Updated belief rules and rule weights of BRB 1 without special behavioral constraints.

Rule number	Updated rule weight	*FlowDiff* and *PressureDiff*	*LeakSize* distribution $\{D_1, D_2, D_3, D_4, D_5\} = \{0, 2, 4, 6, 8\}$
1	1	NL and NL	$\{(D_1, 0), (D_2, 0), (D_3, 0), (D_4, 0), (D_5, 1)\}$
2	1	NL and NM	$\{(D_1, 0), (D_2, 0), (D_3, 0), (D_4, 0.3000), (D_5, 0.7000)\}$
3	1	NL and NS	$\{(D_1, 0.0009), (D_2, 0.0002), (D_3, 0.1990), (D_4, 0.7992), (D_5, 0.0006)\}$
4	0.9937	NL and *Z*	$\{(D_1, 0.0001), (D_2, 0.0020), (D_3, 0.7898), (D_4, 0.2025), (D_5, 0.0057)\}$
5	0.9997	NL and PS	$\{(D_1, 0.6456), (D_2, 0.3505), (D_3, 0.0010), (D_4, 0.0011), (D_5, 0.0018)\}$
6	1	NL and PM	$\{(D_1, 0.8500), (D_2, 0.1500), (D_3, 0), (D_4, 0), (D_5, 0)\}$
7	1	NL and PL	$\{(D_1, 0.9500), (D_2, 0.0500), (D_3, 0), (D_4, 0), (D_5, 0)\}$
8	1	NM and NL	$\{(D_1, 0), (D_2, 0), (D_3, 0.1000), (D_4, 0.9000), (D_5, 0)\}$
9	1	NM and NM	$\{(D_1, 0), (D_2, 0), (D_3, 0..7000), (D_4, 0.3000), (D_5, 0)\}$
10	0.9993	NM and NS	$\{(D_1, 0..0004), (D_2, 0.6977), (D_3, 0.2961), (D_4, 0.0023), (D_5, 0.0036)\}$
11	1	NM and *Z*	$\{(D_1, 0.0011), (D_2, 0.8177), (D_3, 0.0609), (D_4, 0.0492), (D_5, 0.0712)\}$
12	0.9978	NM and PS	$\{(D_1, 0.7779), (D_2, 0.2043), (D_3, 0.0010), (D_4, 0.0063), (D_5, 0.0105)\}$
13	1	NM and PM	$\{(D_1, 0.9000), (D_2, 0.1000), (D_3, 0), (D_4, 0), (D_5, 0)\}$
14	1	NM and PL	$\{(D_1, 0.9900), (D_2, 0.0100), (D_3, 0), (D_4, 0), (D_5, 0)\}$
15	1	NS and NL	$\{(D_1, 0), (D_2, 0), (D_3, 0.4000), (D_4, 0.6000), (D_5, 0)\}$
16	1	NS and NM	$\{(D_1, 0), (D_2, 0), (D_3, 0.8000), (D_4, 0.2000), (D_5, 0)\}$
17	1	NS and NS	$\{(D_1, 0.0011), (D_2, 0.2972), (D_3, 0.5981), (D_4, 0.1016), (D_5, 0.0020)\}$
18	0.9868	NS and *Z*	$\{(D_1, 0.0732), (D_2, 0.5901), (D_3, 0.1951), (D_4, 0.0608), (D_5, 0.0809)\}$
19	0.9971	NS and PS	$\{(D_1, 0.6691), (D_2, 0.2938), (D_3, 0.0024), (D_4, 0.0150), (D_5, 0.0197)\}$
20	1	NS and PM	$\{(D_1, 0.9000), (D_2, 0.1000), (D_3, 0), (D_4, 0), (D_5, 0)\}$
21	1	NS and PL	$\{(D_1, 1), (D_2, 0), (D_3, 0), (D_4, 0), (D_5, 0)\}$
22	1	NVS and NL	$\{(D_1, 0.0200), (D_2, 0.1100), (D_3, 0.3900), (D_4, 0.4800), (D_5, 0)\}$
23	1	NVS and NM	$\{(D_1, 0.1000), (D_2, 0.7800), (D_3, 0.1200), (D_4, 0), (D_5, 0)\}$
24	0.9557	NVS and NS	$\{(D_1, 0.3724), (D_2, 0.5740), (D_3, 0.0133), (D_4, 0.0191), (D_5, 0.0211)\}$
25	0.9838	NVS and *Z*	$\{(D_1, 0.3283), (D_2, 0.2790), (D_3, 0.0664), (D_4, 0.1479), (D_5, 0.1784)\}$
26	0.9921	NVS and PS	$\{(D_1, 0.9092), (D_2, 0.0121), (D_3, 0.0103), (D_4, 0.0306), (D_5, 0.0379)\}$
27	1	NVS and PM	$\{(D_1, 1), (D_2, 0), (D_3, 0), (D_4, 0), (D_5, 0)\}$
28	1	NVS and PL	$\{(D_1, 1), (D_2, 0), (D_3, 0), (D_4, 0), (D_5, 0)\}$
29	1	*Z* and NL	$\{(D_1, 1), (D_2, 0), (D_3, 0), (D_4, 0), (D_5, 0)\}$
30	1	*Z* and NM	$\{(D_1, 1), (D_2, 0), (D_3, 0), (D_4, 0), (D_5, 0)\}$
31	1	*Z* and NS	$\{(D_1, 0.9866), (D_2, 0.0068), (D_3, 0.0019), (D_4, 0.0023), (D_5, 0.0024)\}$
32	0.9999	*Z* and *Z*	$\{(D_1, 0.4303), (D_2, 0.4294), (D_3, 0.0389), (D_4, 0.0489), (D_5, 0.0526)\}$
33	0.994	*Z* and PS	$\{(D_1, 0.9313), (D_2, 0.0345), (D_3, 0.0107), (D_4, 0.0116), (D_5, 0.0118)\}$
34	1	*Z* and PM	$\{(D_1, 1), (D_2, 0), (D_3, 0), (D_4, 0), (D_5, 0)\}$
35	1	*Z* and PL	$\{(D_1, 1), (D_2, 0), (D_3, 0), (D_4, 0), (D_5, 0)\}$
36	1	PS and NL	$\{(D_1, 0.3900), (D_2, 0.6100), (D_3, 0), (D_4, 0), (D_5, 0)\}$
37	1	PS and NM	$\{(D_1, 0.9000), (D_2, 0.1000), (D_3, 0), (D_4, 0), (D_5, 0)\}$
38	1	PS and NS	$\{(D_1, 1), (D_2, 0), (D_3, 0), (D_4, 0), (D_5, 0)\}$
39	0.9995	PS and *Z*	$\{(D_1, 0.9528), (D_2, 0.0354), (D_3, 0.0041), (D_4, 0.0039), (D_5, 0.0038)\}$
40	1	PS and PS	$\{(D_1, 0.9948), (D_2, 0.0021), (D_3, 0.0010), (D_4, 0.0010), (D_5, 0.0010)\}$
41	1	PS and PM	$\{(D_1, 1), (D_2, 0), (D_3, 0), (D_4, 0), (D_5, 0)\}$
42	1	PS and PL	$\{(D_1, 1), (D_2, 0), (D_3, 0), (D_4, 0), (D_5, 0)\}$
43	1	PM and NL	$\{(D_1, 0.1000), (D_2, 0.9000), (D_3, 0), (D_4, 0), (D_5, 0)\}$
44	1	PM and NM	$\{(D_1, 0.3000), (D_2, 0.7000), (D_3, 0), (D_4, 0), (D_5, 0)\}$
45	1	PM and NS	$\{(D_1, 0.8500), (D_2, 0.1500), (D_3, 0), (D_4, 0), (D_5, 0)\}$
46	1	PM and *Z*	$\{(D_1, 0.9800), (D_2, 0.0200), (D_3, 0), (D_4, 0), (D_5, 0)\}$
47	1	PM and PS	$\{(D_1, 1), (D_2, 0), (D_3, 0), (D_4, 0), (D_5, 0)\}$
48	1	PM and PM	$\{(D_1, 1), (D_2, 0), (D_3, 0), (D_4, 0), (D_5, 0)\}$
49	1	PM and PL	$\{(D_1, 1), (D_2, 0), (D_3, 0), (D_4, 0), (D_5, 0)\}$
50	1	PL and NL	$\{(D_1, 0.9000), (D_2, 0.1000), (D_3, 0), (D_4, 0), (D_5, 0)\}$
51	1	PL and NM	$\{(D_1, 0.9900), (D_2, 0.0100), (D_3, 0), (D_4, 0), (D_5, 0)\}$
52	1	PL and NS	$\{(D_1, 1), (D_2, 0), (D_3, 0), (D_4, 0), (D_5, 0)\}$
53	1	PL and *Z*	$\{(D_1, 1), (D_2, 0), (D_3, 0), (D_4, 0), (D_5, 0)\}$
54	1	PL and PS	$\{(D_1, 1), (D_2, 0), (D_3, 0), (D_4, 0), (D_5, 0)\}$
55	1	PL and PM	$\{(D_1, 1), (D_2, 0), (D_3, 0), (D_4, 0), (D_5, 0)\}$
56	1	PL and PL	$\{(D_1, 1), (D_2, 0), (D_3, 0), (D_4, 0), (D_5, 0)\}$

Table 21
Updated belief rules and rule weights of BRB_2 without special behavioral constraints.

Rule number	Updated rule weight	*LeakSize*	*LeakSize_P* distribution $\{D_1, D_2, D_3, D_4, D_5\} = \{0, 2, 4, 6, 8\}$
1	0.9139	D_1	$\{(D_1, 0.1430), (D_2, 0.0650), (D_3, 0.1512), (D_4, 0.2501), (D_5, 0.3907)\}$
2	0.979	D_2	$\{(D_1, 0.0161), (D_2, 0.0728), (D_3, 0.1618), (D_4, 0.2745), (D_5, 0.4748)\}$
3	0.9548	D_3	$\{(D_1, 0.0290), (D_2, 0.0794), (D_3, 0.3330), (D_4, 0.4520), (D_5, 0.1065)\}$
4	0.9295	D_4	$\{(D_1, 0.0577), (D_2, 0.0082), (D_3, 0.0225), (D_4, 0.3386), (D_5, 0.5730)\}$
5	0.916	D_5	$\{(D_1, 0.0557), (D_2, 0.0090), (D_3, 0.0192), (D_4, 0.0315), (D_5, 0.8847)\}$

constraints given in Eqs. (86)–(90) are not considered. The updated BRB_1 and BRB_2 are given in Tables 16 and 17, respectively. Then the last 17 leak datasets are used to test the updated model. Similarly, the mean squared error (MSE) is used [22]. The MSE between the testing data and the forecasting *LeakSize* generated by the initial model is 9.4103. On the other hand, the MSE between the testing data and the forecasting *LeakSize* generated by the updated model without the specific behavioral constraints is 0.0241. Obviously, the MSE generated by the updated forecasting model is far smaller than the initial one. Therefore, the forecasting model composed of the two updated BRBs can also predict the leak size fairly accurately.

However, the rules marked in grey as shown in Tables 20 and 21 do not satisfy the constraints given in Eqs. (86)–(90). For example, in BRB_1 as represented by Table 20, rules 3–4 and rules 24–25 do not satisfy the inequality constraints given in Eq. (86), and rules 10–12 do not satisfy the equality constraints given in Eq. (87). In BRB_2 as represented by Table 21, rules 1–2 and rules 4–5 do not satisfy the constraints given in Eqs. (89) and (90). This means that if the experts can not provide the constraints on the belief degrees to denote the running patterns of oil pipeline leak, the updated model may violate the special running patterns. It may lead to such a result that although the updated model can predict the *LeakSize* accurately for the datasets tested, the belief degrees generated by the updated model are not correct or logical. For example, in rules 10–12 of BRB_1 as represented by Table 20, the belief degrees to the linguistic term D_5 should be zeros according to the analysis of the running patterns of oil pipeline leak, where D_5 denotes that leak is the most severe. In this BRB_1, however, they are not zeros. When these three rules are used to interpret the running condition of oil pipeline, unreasonable or incorrect results may be generated.

On the contrary, the updated belief degrees of the two BRBs as given in Tables 18 and 19 satisfy all these constraints. This ensures that the updated BRBs can generate reasonable results in a reliable and logical way. In general, it is necessary to consider expert intervention in the proposed forecasting model and recursive algorithm.

5. Conclusions

In this paper, based on the concept of belief rule base (BRB), a new forecasting model consisting of two BRBs was constructed. Expert knowledge can be taken into account for training the new models. Two recursive algorithms for updating the forecasting model were proposed with one used to deal with judgmental outputs and the other with numerical outputs. A numerical example and a case study were examined to demonstrate how the new forecasting model is constructed and the proposed recursive algorithms are implemented.

There are several features in the new forecasting model and the proposed algorithms. First of all, different from other optimization models for training BRB systems, the proposed recursive algorithms with an analytic formulation can update the forecasting model once some information is available without having to wait until a complete set of information is all collected, which saves time and is of great practical significance. Secondly, the initial parameters of the forecasting model can be obtained in the following ways: (1) given randomly and (2) manually tuned by experts. If the parameters of the forecasting model are appropriately tuned by experts, the convergence speed of the proposed recursive algorithms can be improved. Thirdly, in the new forecasting model, only partial input and output information is required which could be incomplete, vague, numerical or judgmental. This comes from the unique feature of RIMER [45]. Finally but by no means least importantly, expert knowledge about system behaviors can be imbedded in the training and updating process of the forecasting model, which means that in the proposed recursive algorithms for updating the forecasting model the direct intervention of human experts is allowed. The above features equip the new BRB based forecasting model with the capability of representing a range of real systems, especially when there are strict real-time requirements and uncertainties.

In this paper, the forecasting model is composed of two BRBs, which can be considered as a special hierarchical BRB system. Because a hierarchical BRB system can model more complicated systems, it is important that hierarchical BRB can be updated recursively using newly available information. Based on the study reported in [45,46], the proposed recursive algorithms can be extended to update general hierarchical BRB systems recursively. This means that the proposed algorithms can be used not only to build forecasting models with system behaviors taken into account, but also as a tool for updating hierarchical BRB systems.

Acknowledgements

Zhou thanks the partial support by the NSFC under Grant 61004069 and the Foundation of Department of Education of Ji Lin Province of China under Grant 2009109. Hu thanks the partial support by the NCET under Grant 07144. Xu and Yang thank the partial support by the UK Engineering and Physical Science Research Council under Grant No.: EP/F024606/1 and by the NSFC under Grant 60736026; Zhou thanks the partial support by the National 973 Project under Grants 2010CB731800 and 2009CB32602, and the NSFC under Grants 60721003 and 60736026.

The authors thank the editor and three anonymous referees for their constructive comments and suggestions.

Appendix A. Rule based information transformation technique

In the proposed scheme, there is an important technique, i.e., rule based information transformation technique [44], which is used to transform input information, including qualitative assessment and quantitative data. In this paper, we will use this technique for the quantitative data. So we review it in this Appendix.

Suppose that the input of a quantitative antecedent attribute is given by numerical value. In this case, equivalence rules need to be extracted from the decision maker. This can be used to transform a value to an equivalent expectation, thereby relating a particular value to a set of referential values [45]. Therefore, a value $\gamma_{i,j}$ ($i = 1,\ldots,M$; $j = 1,\ldots,J_i$) can be judged to be a referential value $A_{i,j}$ in BRB, or

$$\gamma_{i,j} \text{ means } A_{i,j}, \quad i = 1,\ldots,M;\ j = 1,\ldots,J_i. \tag{A.1}$$

Suppose that a large value $\gamma_{i,(j+1)}$ is preferred over a small value $\gamma_{i,j}$. Let γ_{i,J_i} and $\gamma_{i,1}$ be the largest and smallest feasible values, respectively. Then, an input value $\hat{x}_i(n)$ is represented using the following equivalent expectation:

$$S(\hat{x}_i(n)) = \left\{ \left(\gamma_{i,j}, \alpha_{i,j}(\hat{x}_i(n)) \right),\ i = 1,\ldots,M;\ j = 1,\ldots,J_i \right\}, \tag{A.2}$$

where $\alpha_{i,j}(\hat{x}_i(n))$ can be calculated by

$$\alpha_{i,j}(\hat{x}_i(n)) = \frac{\gamma_{i,j+1} - \hat{x}_i(n)}{\gamma_{i,j+1} - \gamma_{i,j}} \quad \text{if } \gamma_{i,j} \leqslant \hat{x}_i(n) \leqslant \gamma_{i,j+1},\ j = 1,\ldots,J_i - 1, \tag{A.3}$$

$$\alpha_{i,j+1}(\hat{x}_i(n)) = 1 - \alpha_{i,j}(\hat{x}_i(n)) \quad \text{if } \gamma_{i,j} \leqslant \hat{x}_i(n) \leqslant \gamma_{i,j+1},\ j = 1,\ldots,J_i - 1, \tag{A.4}$$

$$\alpha_{i,s}(\hat{x}_i(n)) = 0 \quad \text{for } s = 1,\ldots,J_i,\ s \neq j, j+1. \tag{A.5}$$

A quantitative antecedent attribute, $\hat{x}_i(n)$, may also be a random variable and may not always take a single value but several values with different probabilities. In order to solve this problem, the corresponding rule based information transformation technique has also been proposed [44].

Appendix B. Calculation of the derivatives

In Eqs. (32), (33), (42) and (43), the derivatives of β_m ($m = 1,\ldots,N$) with respect to parameter vector $\mathbf{V}$ are used, and they are given in this Appendix.

When $(n + p)$ is shaded, Eq. (27) is written as:

$$f\left(\widehat{\mathbf{B}} \middle| \mathbf{Q}\right) = (2\pi)^{-N/2} |\boldsymbol{\chi}|^{-1/2} \exp\left\{ -\frac{1}{2} (\hat{\mathbf{B}} - \mathbf{B})^T \boldsymbol{\chi}^{-1} (\hat{\mathbf{B}} - \mathbf{B}) \right\}, \tag{B.1}$$

$$\log f\left(\widehat{\mathbf{B}} \middle| \mathbf{Q}\right) = -\frac{N}{2} \log 2\pi - \frac{1}{2} \log |\boldsymbol{\chi}| \frac{1}{2} (\widehat{B} - \mathbf{B})^T \boldsymbol{\chi}^{-1} (\widehat{\mathbf{B}} - \mathbf{B}), \tag{B.2}$$

where $\widehat{\mathbf{B}} = \left[\hat{\beta}_1,\ldots,\hat{\beta}_N\right]^T$ and $\mathbf{B} = [\beta_1,\ldots,\beta_N]^T$.

The first derivatives of $\log f\left(\widehat{\mathbf{B}} \middle| \mathbf{Q}\right)$ with respect to V_a ($a = 1,\ldots,L + M + L \times N + N + N \times N$) are:

$$\frac{\partial \log f\left(\widehat{\mathbf{B}} \middle| \mathbf{Q}\right)}{\partial V_a} = \frac{\partial \mathbf{B}^T}{\partial V_a} \boldsymbol{\chi}^{-1} (\widehat{\mathbf{B}} - \mathbf{B}), \tag{B.3}$$

where $\frac{\partial \beta_j}{\partial V_a}$ needs to be calculated. $\mathbf{V} = \left[\theta_k^1, \bar{\delta}_m^1, \beta_{j,k}^1, \theta_l^2, \beta_{i,l}^2\right]^T$ and $k = 1,\ldots,L$, $m = 1,\ldots,M$, $j = 1,\ldots,N$, $l = 1,\ldots,N$ and $i = 1,\ldots,N$.

According to the ER analytical algorithm, the derivatives of β_m ($m = 1,\ldots,N$) with respect to the parameters of BRB_1 can be written as

$$\begin{cases} \frac{\partial \beta_m}{\partial \theta_s^1} = \frac{\partial \beta_m}{\partial \omega_l^2} \frac{\partial \omega_l^2}{\partial \beta_j^1} \frac{\partial \beta_j^1}{\partial \theta_s^1}, & j = 1,\ldots,N,\ l = 1,\ldots,N,\ s = 1,\ldots,L, \\ \frac{\partial \beta_m}{\partial \bar{\delta}_i^1} = \frac{\partial \beta_m}{\partial \omega_l^2} \frac{\partial \omega_l^2}{\partial \beta_j^1} \frac{\partial \beta_j^1}{\partial \bar{\delta}_i^1}, & j = 1,\ldots,N,\ l = 1,\ldots,N,\ i = 1,\ldots,M, \\ \frac{\partial \beta_m}{\partial \beta_{z,q}^1} = \frac{\partial \beta_m}{\partial \omega_l^2} \frac{\partial \omega_l^2}{\partial \beta_j^1} \frac{\partial \beta_j^1}{\partial \beta_{z,q}^1} & j = 1,\ldots,N,\ l = 1,\ldots,N,\ z = 1,\ldots,N,\ q = 1,\ldots,L, \end{cases} \tag{B.4}$$

where ω_l^2 is the activation weight of BRB_2. $\frac{\partial \beta_m}{\partial \omega_l^2}$, $\frac{\partial \beta_j^1}{\partial \theta_s^1}$, $\frac{\partial \beta_j^1}{\partial \bar{\delta}_i^1}$ and $\frac{\partial \beta_j^1}{\partial \beta_{z,q}^1}$ are given in [59]. $\frac{\partial \omega_l^2}{\partial \beta_j^1}$ is calculated by

$$\frac{\partial \omega_l^2}{\partial \beta_j^1} = \begin{cases} -\dfrac{\theta_l^2 \theta_j^2 \bar{\delta}_1^2 \left(\beta_l^1\right)^{\bar{\delta}_1^2} \left(\beta_j^1\right)^{\bar{\delta}_1^2 - 1}}{\left[\sum_{k=1}^N \theta_k \left(\beta_k^1\right)^{\bar{\delta}_1^2}\right]^2}, & j \neq l, \\ \dfrac{\theta_j^2 \bar{\delta}_1^2 \left(\beta_j^1\right)^{\bar{\delta}_1^2 - 1}}{\sum_{k=1}^N \theta_k \left(\beta_k^1\right)^{\bar{\delta}_1^2}} - \dfrac{\theta_l^2 \theta_j^2 \bar{\delta}_1^2 \left(\beta_l^1\right)^{\bar{\delta}_1^2} \left(\beta_j^1\right)^{\bar{\delta}_1^2 - 1}}{\left[\sum_{k=1}^N \theta_k \left(\beta_k^1\right)^{\bar{\delta}_1^2}\right]^2}, & j = l. \end{cases} \tag{B.5}$$

The derivatives of β_m with respect to the parameters of the BRB_2 can be also been given as in [59].

References

[1] C. Angeli, A. Chatzinikolaou, Prediction and diagnosis of faults in hydraulic systems, Proceedings of the Institution of Mechanical Engineers 216 (2002) 293–297.

[2] P.P. Balestrassi, E. Popova, A.P. Paiva, J.W. Marangon Lima, Design of experiments on neural network's training for nonlinear time series forecasting, Neurocomputing 72 (2009) 1160–1178.

[3] T.G. Barbounis, J.B. Theocharis, Locally recurrent neural networks for wind speed prediction using spatial correlation, Information Sciences 177 (2007) 5775–5797.

[4] G. Cardoso, F. Gomide, Newspaper demand prediction and replacement model based on fuzzy clustering and rules, Information Sciences 177 (2007) 4799–4809.

[5] K.Y. Chen, Forecasting system's reliability based on support vector regression with genetic algorithms, Reliability Engineering and System Safety 92 (2007) 423–432.

[6] S.M. Chen, R.H. Jeng, Temperature prediction using fuzzy time series, IEEE Transactions on Systems, Man, and Cybernetics (Part B) 30 (2000) 363–371.

[7] W.T. Chen, M. Saif, A novel fuzzy system with dynamic rule base, IEEE Transactions on Fuzzy Systems 13 (2005) 569–582.

[8] M.Z. Chen, D.H. Zhou, G.P. Liu, A new particle predictor for fault prediction of nonlinear time-varying systems, Developments in Chemical and Engineering and Mineral Processing 13 (2005) 379–388.

[9] J.L. Cheng, C.C. Cheng, Prediction and identification using wavelet-based recurrent fuzzy neural networks, IEEE Transactions on Systems, Man, and Cybernetics (Part B) 34 (2004) 2144–2154.

[10] P.J. Chung, J.F. Bohme, Recursive EM and SAGE-inspired algorithms with application to DOA estimation, IEEE Transactions on Signal Processing 53 (2005) 2664–2677.

[11] A.P. Dempster, Upper and lower probabilities induced by a multi-valued mapping, Annals Mathematical Statistics 38 (1967) 325–339.

[12] A.P. Dempster, A generalization of Bayesian inference, Journal of the Royal Statistical Society (Series B) 30 (1968) 205–247.

[13] A.P. Dempster, N. Laird, D.B. Rubin, Maximum likelihood from incomplete data via the EM algorithm, Journal of the Royal Statistical Society (Series B) 39 (1977) 1–38.

[14] J.G.D. Gooijer, R.J. Hyndman, 25 years of time series forecasting, International Journal of Forecasting 22 (2006) 443–473.

[15] C. Hamzacebi, Improving artificial neural networks' performance in seasonal time series forecasting, Information Sciences 178 (2008) 4550–4559.

[16] S.L. Ho, M. Xie, The use of ARIMA models for reliability forecasting and analysis, Computers and Industry Engineering 35 (1998) 213–216.

[17] C.L. Huang, K. Yong, Multiple Attribute Decision Making Methods and Applications, A State-of Art Survey, Springer-Verlag, Berlin, 1981.

[18] Y.Q. Jiang, Applying grey forecasting to predicting the operating energy performance of air cooled water chillers, International Journal of Refrigeration 27 (2004) 385–392.

[19] H.J. Kushner, M.L. Kelmanson, Stochastic approximation algorithms of the multiplier type for the sequential Monte Carlo optimization of stochastic systems, SIAM Journal of Control and Optimization 14 (1976) 827–842.

[20] H.J. Kushner, G.G. Yin, Stochastic Approximation Algorithms and Applications, Springer-Verlag, New York, 1997.

[21] H.K. Kwan, Y.L. Cai, A fuzzy neural-network and its application to pattern recognition, IEEE Transactions on Fuzzy Systems 2 (1994) 185–193.

[22] C.D. Lewis, Industrial and Business Forecasting Method, Butterworth Scientific, 1982.

[23] W.H. Lee, S.S. Tseng, S.H. Tsai, A knowledge based real-time travel time prediction system for urban network, Expert Systems with Applications 36 (2009) 4239–4247.

[24] S.T. Li, S.C. Kuo, Y.C. Cheng, C.C. Chen, Deterministic vector long-term forecasting for fuzzy time series, Fuzzy Sets and Systems 161 (2010) 1852–1870.

[25] H. Li, J. Sun, Gaussian case-based reasoning for business failure prediction with empirical data in China, Information Sciences 179 (2009) 89–108.

[26] J. Liu, J.B. Yang, H.S. Sii, Engineering system safety analysis and synthesis using fuzzy rule based evidential reasoning approach, Quality and Reliability Engineering International 21 (2005) 387–411.

[27] J. Liu, J.B. Yang, H.S. Sii, Y.M. Wang, Fuzzy rule-based evidential reasoning approach for safety analysis, International Journal of General Systems 33 (2004) 183–204.

[28] H.T. Lu, W.J. Kolarik, S.S. Lu, Real-time performance reliability prediction, IEEE Transactions on Reliability 50 (2004) 353–357.

[29] J. Pearl, Probabilistic Reasoning in Intelligence Systems, Morgan Kaufman, San Mateo, CA, 1988.

[30] H.T. Pham, B.S. Yang, Estimation and forecasting of machine health condition using ARMA/GARCH model, Mechanical Systems and Signal Processing 24 (2010) 546–558.

[31] M. Setnes, R. Babuska, U. Kaymak, H.R. Van Nauta Lemke, Similarity measures in fuzzy rule base simplification, IEEE Transactions on Systems, Man, and Cybernetics (Part B) 37 (1988) 376–386.

[32] G. Shafer, A Mathematical Theory of Evidence, Princeton University Press, Princeton, NJ, 1976.

[33] C.G. Silva, Time series forecasting with a non-linear model and the scatter search meta-heuristic, Information Sciences 178 (2008) 3288–3299.

[34] C. Simon, P. Weber, A. Evsukoff, Bayesian networks inference algorithm to implement Dempster–Shafer theory in reliability analysis, Reliability Engineering and System Safety 93 (2008) 950–963.

[35] D.J. Spiegelhalter, A.P. Dawld, S.L. Lauritzen, R.G. Cowell, Bayesian analysis in expert systems, Statistical Science 8 (1993) 219–247.

[36] R. Sun, Robust reasoning: integrating rule-based and similarity-based reasoning, Artificial Intelligence 75 (1995) 241–295.

[37] Z.L. Sun, K.F. Au, T.M. Chos, A neuron-fuzzy inference system through integration of fuzzy logic and extreme learning machines, IEEE Transactions on Systems, Man, and Cybernetics (Part B) 37 (2007) 1321–1331.

[38] D.M. Titterington, Recursive parameter estimation using incomplete data, Journal of the Royal Statistical Society (Series B) 46 (1984) 257–267.

[39] P. Walley, Measures of uncertainty in expert system, Artificial Intelligence 83 (1996) 1–58.

[40] Y.M. Wang, J.B. Yang, D.L. Xu, Environmental impact assessment using the evidential reasoning approach, European Journal of Operational Research 174 (2006) 1885–1913.

[41] D. Wang, D.H. Zhou, Y.H. Jin, S.J. Qin, A strong tracking predictor for nonlinear processes with input time delay, Computers and Chemical Engineering 28 (2004) 2523–2540.

[42] D.L. Xu, J. Liu, J.B. Yang, G.P. Liu, J. Wang, I. Jenkinson, J. Ren, Inference and learning methodology of belief-rule-based expert system for pipeline leak detection, Expert Systems with Applications 32 (2007) 103–113.

[43] R.R. Yager, OWA aggregation over a continuous interval argument with application to decision making, IEEE Transactions on Systems, Man, and Cybernetics (Part B) 34 (5) (2004) 1952–1963.

[44] J.B. Yang, Rule and utility based evidential reasoning approach for multi-attribute decision analysis under uncertainties, European Journal of Operational Research 131 (2001) 31–61.

[45] J.B. Yang, J. Liu, J. Wang, H.S. Sii, H.W. Wang, Belief rule-base inference methodology using the evidential reasoning approach – RIMER, IEEE Transactions on Systems, Man, and Cybernetics (Part A) 36 (2006) 266–285.

[46] J.B. Yang, J. Liu, D.L. Xu, J. Wang, H.W. Wang, Optimal learning method for training belief rule based systems, IEEE Transactions on Systems, Man, and Cybernetics (Part A) 37 (2007) 569–585.

[47] J.B. Yang, P. Sen, A general multi-level evaluation process for hybrid MADM with uncertainty, IEEE Transactions on Systems, Man, and Cybernetics (Part A) 24 (1994) 1458–1473.

[48] J.B. Yang, M.G. Singh, An evidential reasoning approach for multiple attribute decision making with uncertainty, IEEE Transactions on Systems, Man, and Cybernetics (Part A) 24 (1994) 1–18.

[49] J.B. Yang, Y.M. Wang, D.L. Xu, K.S. Chin, The evidential reasoning approach for MADA under both probabilistic and fuzzy uncertainties, European Journal of Operational Research 171 (2006) 309–343.

[50] J.B. Yang, D.L. Xu, Intelligent Decision System Via Evidential Reasoning – Version 1.1, IDSL, Cheshire, UK, 1999.
[51] J.B. Yang, D.L. Xu, On the evidential reasoning algorithm for multiple attribute decision analysis under uncertainty, IEEE Transactions on Systems, Man, and Cybernetics (Part A) 32 (2002) 289–304.
[52] J.B. Yang, D.L. Xu, Nonlinear information aggregation via evidential reasoning in multi-attribute decision analysis under uncertainty, IEEE Transactions on Systems, Man, and Cybernetics (Part A) 32 (2002) 376–393.
[53] S.K. Yang, T.S. Liu, A Petri net approach to early failure detection and isolation for preventive maintenance, Quality and Reliability Engineering International 14 (1998) 319–330.
[54] S.K. Yang, T.S. Liu, State estimation for predictive maintenance using Kalman filter, Reliability Engineering and System Safety 66 (1999) 29–39.
[55] L.Z. Zadeh, Fuzzy sets, Information and Control 8 (1965) 338–353.
[56] Z.J. Zhou, C.H. Hu, An effective hybrid approach based on grey and ARMA for forecasting gyro drift, Chaos, Solitons & Fractals 35 (2008) 525–529.
[57] Z.J. Zhou, C.H. Hu, H.D. Fan, J. Li, Fault prediction of the nonlinear systems with uncertainty, Simulation Modelling Practice and Theory 16 (2008) 690–703.
[58] Z.J. Zhou, C.H. Hu, J.B. Yang, D.L. Xu, D.H. Zhou, Online updating belief rule based system for pipeline leak detection under expert intervention, Expert Systems with Applications 36 (2009) 7700–7709.
[59] Z.J. Zhou, C.H. Hu, J.B. Yang, D.L. Xu, D.H. Zhou, Online updating belief-rule-based systems using the RIMER approach, IEEE Transactions on Systems, Man, and Cybernetics – Part A: Systems and Humans, in press.
[60] Z.J. Zhou, C.H. Hu, J.B. Yang, D.L. Xu, M.Y. Chen, D.H. Zhou, A sequential learning algorithm for online constructing belief-rule-based systems, Expert Systems with Applications 37 (2010) 1790–1799.
[61] Z.J. Zhou, C.H. Hu, D.H. Zhou, Fault prediction techniques for dynamic systems based on non-analytical model, Information and Control 35 (2006) 608–613 (in Chinese).