Neural Comput & Applic
DOI 10.1007/s00521-016-2341-5

ORIGINAL ARTICLE

Dynamic behavioral assessment model based on Hebb learning rule

Yunfei Yin[1,2,3] · Hailong Yuan[2] · Beilei Zhang[2]

Received: 5 October 2015 / Accepted: 9 May 2016

Abstract Behavioral assessment based on computing system is with important value for computer-simulated training and system diagnosis. However, the existing assessment is a static method for ex post evaluation, and the low efficiency and high complexity have been the urgent problems to be solved in the academic field. In this paper, we propose an adaptive dynamic behavioral assessment model based on Hebb learning rule that effectively combines the assessment standard and the weights of factors. The dynamic behavioral assessment considers the relative weights between the assessment indexes, whereas the existing assessment method does not; the dynamic behavioral assessment uses the assessment standard data recursively and can conduct an instant assessment for the objectives. We have built an assessment system for computer-simulated training, and took the pilot behavioral assessment for example to test and verify the dynamic behavioral assessment mode. Experimental results show that the dynamic behavioral assessment model based on Hebb learning rule has more advantage in assessment efficiency and online computing support.

✉ Yunfei Yin
yinyunfei@cqu.edu.cn

1 Key Laboratory of Dependable Service Computing in Cyber Physical Society (Chongqing University), Ministry of Education, Chongqing, China

2 College of Computer Science, Chongqing University, Chongqing 400044, China

3 Jiangsu Key Laboratory of Meteorological Observation and Information Processing, Nanjing University of Information Science and Technology, Nanjing, China

Keywords Dynamic assessment · Hebb learning · Behavioral data · Adaptive

1 Introduction

Behavioral assessment is one of the important steps for human to understand the world, and plays a decisive role in the research of performance appraisal, system diagnosis, simulation training and machine learning. Behavioral assessment is defined as "using the preset behavior to evaluate the actual behavior". If the behavior is simple and denoted by single-element variable, the assessment process is also relative simple; if the behavior is complex and denoted by multiple elements variable, the assessment process also becomes complex. Therefore, as to the behavioral assessment, the research difficult is the complex behavior assessment of multiple elements variable, while the research hot spot of multiple elements variable behavioral assessment is the determination of relative weights of all the variables. The existing methods for determining the relative weights of multiple elements variable are expert assessment method [1, 2], experience formula method [3, 4], mean–variance method [5, 6], and support–confidence method [7, 8].

Expert assessment method is the method to conduct complex behavior assessment by expert scoring, whose accuracy is affected by the experience and the knowledge range and quality of the experts [1, 2]. Therefore, the expert assessment method is use simple and intuitive, but it is with the drawbacks of too strong subjectivity, low assessment efficiency, lack theory, and insufficient systemization. Wang and Durugbo [9] proposed an industrial product–service expert assessment method, which quantized the expert experience and enhanced the assessment

Published online: 18 May 2016

efficiency; Duque-Ramos et al. [10] proposed an OQuaRE assessment framework based on ontology used to evaluate the software aging, which expanded Wang's model.

Experience formula method is the one summarizing formula from the research object and the assessment experience. Commonly, fitting method is adopted to fit the one-variable evaluation function or multiple-variable evaluation function from the experience data [3, 4], and Tansel et al. [11] proposed an experience assessment formula based on machine learning used to assess the communication behavior for internet of things, which fundamentally broke the research framework fitting experience formula. Lai and Jiang [12] proposed a behavioral assessment model based on genetic algorithm used in the research of intelligent planning, and it is a more influential research.

Mean–variance method for assessment is the one that compares the mean and variance of a group of assessment results with the preset mean threshold and variance threshold: if not larger than the preset mean threshold and the variance threshold, the assessment is passed, or the assessment fails [5, 6]. The advantage of this method is the relatively higher efficiency, and the drawback is the lower accuracy. Arroyo and Fernandez [13] improved the mean–variance assessment method, and proposed the *K*-means fuzzy assessment method that improved the accuracy of mean–variance assessment method; Luque-Baena et al. [14] also proposed the fuzzy assessment method based on *K*-means and used it to evaluate the robot behavior.

Support–confidence method is the one assessing system behavior by the concept of support and confidence, and this method is usually combined with the research of data mining [7, 8, 15–17]. Zhou et al. [18] investigated the architecture and behavioral law of internetware and proposed the internetware behavioral assessment method based on support–confidence method, and this is a novel research direction for complex behavioral assessment based on support–confidence method.

However, the existing system is more complex and the running efficiency is lower, and therefore, we cannot conduct dynamic behavioral assessment.

In this research, we take the pilot behavioral assessment, for example, and design an adaptive method to compute relative weight that combines the standard flight data. The model is denoted by $w^{\text{new}} = w^{\text{old}} + \alpha F(X)G(Y)$, where w^{new} denotes the weight in the next time, w^{old} denotes the current weight, α denotes the learning speed, and $F(X)$ and $G(Y)$ denote the standard flight data and the current output data of system, respectively. This model originates from the Hebb learning thought in the research of neural network [19–21], and therefore, it has certain adaptability, and can be used in dynamic behavior assessment. The method proposed in this paper is with the following advantages: (1) method simple, can be easily implemented in computer systems; (2) high assessment efficiency; (3) supporting dynamic assessment; (4) assessment curve generation in real time; (5) assessment combining the standard behavioral data and the relative weights computation.

Among the current methods, the expert assessment method has the highest accuracy compared with the experience formula method, the mean–variance assessment method, and the support–confidence method and therefore is also the most commonly used method. There are many models for expert assessment method, where G1 behavior sequence model [22] is the most mature model, and is commonly used to be the standard to evaluate the good and bad of other models. In this paper, we will compare with the G1 behavior sequence model.

In the next section, we will discuss the behavior assessment model; in Sect. 3, we will discuss the behavior assessment model based on Hebb learning rule; in Sect. 4, we will conduct experimental verification for the behavior assessment model based on Hebb learning rule, including experiment design, experimental results and experiment analysis; Sect. 5 concludes and outlines directions for future research.

2 Behavioral assessment system

Behavioral assessment system is the model based on computer systems to evaluate whether the behavior meets the preset standard or not. In behavior assessment system, the most important is to divide the behavioral process into multiple phases, to follow up, select suitable assessment objectives for the features in each phase, and then, determine the weights of all assessment objectives according to the importance of each objective, and finally, conduct the construction of mathematical model and computer systems, thus getting the scores of behavior assessment.

The behavior assessment model complies with the following definitions.

Definition 1 If the assessment objective set $\{x_1, x_2, x_3, \ldots, x_s\}$ has the relational expression by certain assessment criterion:

$$x_1^* > x_2^* > x_3^* > \cdots > x_s^* \tag{1}$$

So, we call the assessment objective set $\{x_1, x_2, x_3, \ldots, x_s\}$ to have established the ordering relation according to "$>$", where x_i* denotes the ith assessment objective after reordering the $\{x_i\}$ according to the relation "$>$" $(i = 1, 2, \ldots, s)$.

Definition 2 If the assessment weights for the assessment objectives $x_1, x_2, \ldots, x_s$ is $w_1, w_2, \ldots, w_s$, then the total scoring is

$$\begin{cases} w_1x_1 + w_2x_2 + \cdots + w_sx_s \\ \sum_{k=1}^{s} w_k = 1 \end{cases} \quad (2)$$

where $w_1x_1 + w_2x_2 + \cdots + w_sx_s$ denotes the system synthetic scoring based on the assessment objectives $x_1, x_2, \ldots, x_s$.

The commonly used behavior assessment models in engineering include Bayes assessment model [23] and G1 behavior sequence model [22], and so forth. Bayes assessment model belongs to the mean–variance assessment method, and G1 behavior sequence model belongs to the expert assessment method. G1 behavior sequence model is the better in assessment accuracy and has the highest assessment efficiency, therefore is also the most commonly used behavior assessment model. In this study, we select G1 behavior sequence model as the reference and comparison.

G1 behavior sequence model is a sort of subjective weighting method based on "functionality drive". G1 behavior sequence algorithm is divided into three steps.

2.1 Determine ordering relation

The method of determining ordering relation is: Firstly select the most important objectives from the assessment objective set $\{x_1, x_2, x_3, \ldots, x_s\}$ and denote by x_1*; to follow up, select the most important objective in the rest of the $s - 1$ objectives and denote by x_2*; in this analogy, by s selections, the final remaining assessment objective is denoted by x_s*.

2.2 Judge the relative important degree

We judge the relative importance between the assessment objective x_{k-1}* and x_k* and denote it to be $r_k = w_{k-1}/w_k$ which expresses the relative importance between the assessment objective x_{k-1}* and x_k*, where w_{k-1} and w_k are the weights of x_{k-1}* and x_k*, respectively. Obviously, $r_k > 1$; if k is larger, then x_{k-1}* and x_k* are all unimportant objectives, and in that case, $r_k \approx 1$. In G1 behavior sequence algorithm, the taken values of r_k include 10 cases, see also Table 1 for details.

In Table 1, there are 10 taken values for r_k, which corresponds with 10 cases.

Remarks We take $r_k = 1$ if two comparing objects have the same importance and we have greater 90 % confidence for assessment; we take $r_k = 1.1$ if the comparing objects have the same importance and we have less confidence for assessment; we take $r_k = 1.2$ if the comparing object x_{k-1}* is slighter important than x_k* and we have greater 90 % confidence for assessment; we take $r_k = 1.3$ if the comparing object x_{k-1}* is slighter important than x_k* and we have less confidence for assessment. For the rest values of r_k, they can be deduced by analogy.

2.3 Compute the coefficient for weighting

The computational method for the weighting coefficient is a recursive process, viz., firstly determine the weight w_s of objective x_s*, and then successively compute $w_{s-1}, w_{s-2}, \ldots, w_1$, and the corresponding recursive formula is $w_{k-1} = r_k w_k$, where $k = s, s - 1, \ldots, 2$.

Computation for w_s: Since $\sum_{k=1}^{s} w_k = 1$, viz., $w_1 + w_2 + \cdots + w_s = 1$, the equality is divided by w_s, viz., $\frac{w_1}{w_s} + \frac{w_2}{w_s} + \cdots + \frac{w_s}{w_s} = \frac{1}{w_s}$, arrange it and can get $1 + \sum_{i=2}^{s} \frac{w_{i-1}}{w_s} = \frac{1}{w_s}$. Considering $\frac{w_{i-1}}{w_s} = \frac{w_{i-1}}{w_i}\frac{w_i}{w_{i+1}}\cdots\frac{w_{s-1}}{w_s}$, therefore,

$$w_s = \left(1 + \sum_{i=2}^{s}\prod_{j=i}^{s}\frac{w_{j-1}}{w_j}\right)^{-1} = \left(1 + \sum_{i=2}^{s}\prod_{j=i}^{s} r_j\right)^{-1} \quad (3)$$

where r_j takes values according to Table 1.

G1 behavior sequence algorithm is with the advantages of high computational accuracy, unlimited number of assessment objectives, and so on.

Table 1 Taken values of relative important degree

r_k	Descriptions for the taken values	
	Comparison between objective x_{k-1}* and x_k*	Confidence for assessment
1.0	Both of them have the same importance	[0.9, 1]
1.1		[0, 0.9]
1.2	The former is slight important than the latter	[0.9, 1]
1.3		[0, 0.9]
1.4	The former is obviously important than the latter	[0.9, 1]
1.5		[0, 0.9]
1.6	The former is strongly important than the latter	[0.9, 1]
1.7		[0, 0.9]
1.8	The former is extremely important than the latter	[0.9, 1]
1.9		[0, 0.9]

3 Hebb learning rule-based dynamic behavior assessment system

In this section, we firstly introduce the principle of Hebb learning rule, and then propose the basic assumption and design the assessment model and algorithms based on Hebb learning rules.

3.1 Hebb learning rule

Hebb learning rule is a sort of unsupervised learning method [19], and it determines the link weight of neural network according to the current input and output of system, as shown in Fig. 1.

In Fig. 1, $x_1, x_2, \ldots, x_s$ are input variables, $y_1, y_2, \ldots, y_t$ are output variables, and $w_1, w_2, \ldots, w_u$ are the link weights of neural network. In the neural network model, the weights between input and output represent the relationship between input and output. The objective of the system optimization is to find the optimal weights between inputs and outputs by the iterative method.

According to Hebb learning rule, if $x_1, x_2, \ldots, x_s > 0$ and $y_1, y_2, \ldots, y_t > 0$, $w_1, w_2, \ldots, w_u$ have the tendency to increase, and meet:

$$\begin{cases} w_{ij}^{\text{new}} = w_{ij}^{\text{old}} + \Delta w_{ij} \\ \Delta w_{ij} = \alpha f_i(x_1, x_2, \ldots, x_s) g_j(y_1, y_2, \ldots, y_t) \end{cases} \quad (4)$$

In formula (4), w_{ij}^{new} denotes the new weight between the ith input and the jth output, w_{ij}^{old} denotes the old weight between the ith input and the jth output, Δw_{ij} denotes the increment of weight, α denotes the learning speed and takes positive, f_i $(x_1, x_2, \ldots, x_s)$ denotes the composite function of the input $x_1, x_2, \ldots, x_s$, and g_j $(y_1, y_2, \ldots, y_t)$ denotes the composite function of the output $y_1, y_2, \ldots, y_t$.

Hebb learning rule originates from the activity law of neuron, viz., if the two neurons on the two sides of one synapse are activated at the same time, the strength of the synapse will be increased. We use the input and output to denote the two neurons on the two sides of one synapse, and use link weight to denote the link strength of the synapse. When the input and output take positive numbers, we regard them to be activated and the link weight will be increased.

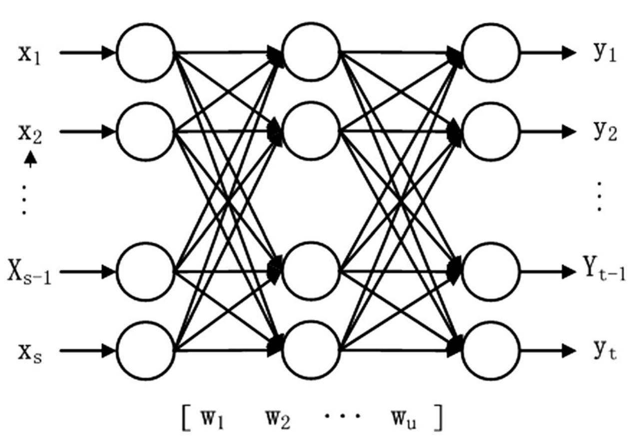


Fig. 1 Model of neural network

Ratiocination 1 When $x_1, x_2, \ldots, x_s < 0$ and $y_1, y_2, \ldots, y_t > 0$, $w_1, w_2, \ldots, w_u$ will have the tendency to decrease.

Proof Suppose $x_1, x_2, \ldots, x_s$ to be the neurons on the left side of the synapse, and $y_1, y_2, \ldots, y_t$ to be the neurons on the right side of the synapse. Since $x_1, x_2, \ldots, x_s < 0$ denotes the left neuron to be activated in the negative direction, and $y_1, y_2, \ldots, y_t > 0$ denotes the right neuron to be activated in the positive direction, both of the information transmission directions are on the contrary. Therefore, both of the link strength will be slacked down, viz., $w_1, w_2, \ldots, w_u$ have the tendency to decrease.

Ratiocination 2 When $x_1, x_2, \ldots, x_s > 0$ and $y_1, y_2, \ldots, y_t < 0$, $w_1, w_2, \ldots, w_u$ will have the tendency to decrease.

Proof Same as the Ratiocination 1.

Ratiocination 3 When $x_1, x_2, \ldots, x_s < 0$ and $y_1, y_2, \ldots, y_t < 0$, $w_1, w_2, \ldots, w_u$ are all equal to 0.

Proof It denotes that the neurons on the left and right sides are all activated in the negative direction for $x_1, x_2, \ldots, x_s < 0$ and $y_1, y_2, \ldots, y_t < 0$, so the link strength is extremely slacked down. In this study, we regard that there are no information transmission between the neurons, viz., $w_1, w_2, \ldots, w_u$ are all equal to 0.

The advantages of Hebb learning rule are: (1) high learning efficiency; (2) less system dependency.

3.2 Hebb dynamic behavioral assessment model

3.2.1 Basic assumptions

The dynamic behavioral assessment model based on Hebb learning rule is based on the following assumptions:

1. System behavioral data is treated with the input data.
2. Assessment result for the system behavioral data is treated with the output data.
3. All the objectives of system behavioral data are independent of each other.
4. The initial link weights between the neurons are designated in advance.

Among the basic assumptions, the assumption (1) is used to limit the form of system input data, viz., the behavioral data of system; the behavioral data of system include the current position of system, attitude, speed, acceleration, and so forth.

The assumption (2) is used to limit the form of system output data, viz., the assessment result of system behavior; the assessment result of system behavior denotes the difference between the system current behavior and the preset standard behavior. We adopt the "max–min error" method to compute the difference between the system current behavior and the preset standard behavior, whose basic idea is as follows.

Suppose the ith objective of behavioral data to be x_i, and let the standard value of x_i be x_i^0, and the minimal and maximal error thresholds of x_i be [ε^-, ε^+], and then the assessment result for x_i is:

$$y_i = \begin{cases} 1 & \left|x_i - x_i^0\right| < \varepsilon^- \\ \dfrac{\varepsilon^+ - \left|x_i - x_i^0\right|}{\varepsilon^+ - \varepsilon^-} \times 100\,\% & \varepsilon^- \le \left|x_i - x_i^0\right| < \varepsilon^+ \\ 0 & \left|x_i - x_i^0\right| \ge \varepsilon^+ \end{cases} \tag{5}$$

Assumption (3) is used to limit each of the objectives of the behavioral data not to affect each other.

Assumption (4) is used to provide the method for taking the initial link weights between neurons, viz., it can take values according to the experience. In our work, the initial link weight between neurons is computed by G1 behavior sequence algorithm designed in Sect. 2.

3.2.2 Model design

According to the basic assumptions, we can design the following training set:

$$\text{TrainSet} = \left\{\left\langle \text{TSI}(x_i), \text{TSO}(y_j)\right\rangle, \text{TR}(w_k); i = 1,2,\ldots,s_T, \; j = 1,2,\ldots,t_T, k = 1,2,\ldots,u_T\right\}$$

where TSI(x_i) denotes the input sequence of training set, which is the system standard behavioral data (known in advance); TSO(y_j) denotes the output sequence of training set, which is the assessment for the system standard behavioral data (known in advance); TR(w_k) is the training result based on Hebb learning rule, which denotes the associated degree between the input sequence and the output sequence; s_T and t_T denote the size of the input sequence and output sequence, respectively; u_T denotes the number of associations taken place between the input sequence and the output sequence.

It is well-known that the training set is necessary for behavioral assessment model, and how to obtain the training data is an important task. In this paper, we have built an assessment system for computer-simulated training, and invited more than 20 pilots from Beijing University of Aeronautics and astronautics to train. The assessment system for computer-simulated training will automatically collect training data, and we put these data as the training set.

Similarly, we design the testing set as follows:

$$\text{EvaluateSet} = \left\{\langle \text{ESI}(x_i), \text{ER}(w_k)\rangle, \text{ESO}(y_j); i = 1,2,\ldots,s_E, \; j = 1,2,\ldots,t_E, k = 1,2,\ldots,u_E\right\}$$

where ESI(x_i) denotes the input sequence of testing set; ER(w_k) denotes the associated degree between the input sequence and the output sequence; ESO(y_j) denotes the output result required to be solved, which is the dynamic assessment result of all the system behavior objectives. s_E and t_E denote the input sequence of testing set and the output sequence of testing set, respectively; u_E denotes the number of associations taken place between the input sequence and the output sequence.

Based on the training set and testing set, we design the Hebb dynamic behavior assessment model as follows.

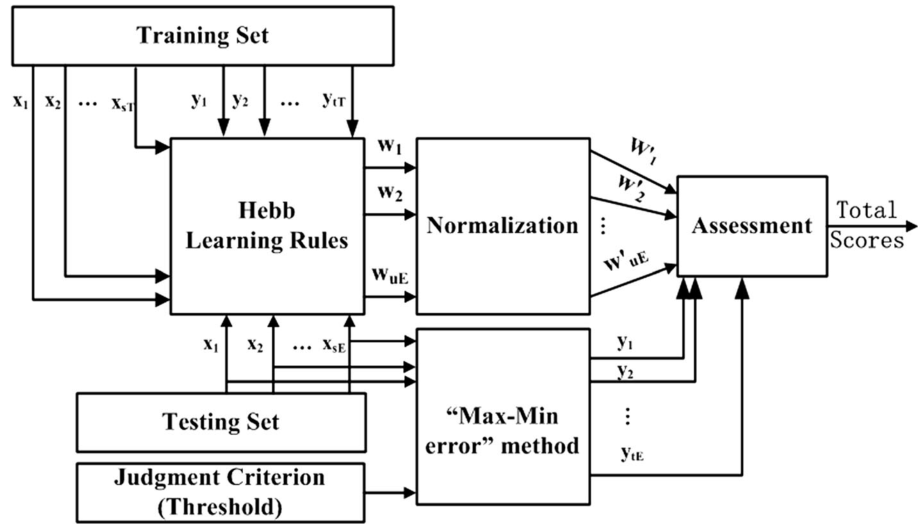
In Fig. 2, the training set provides the input sequence x_1, x_2, …, x_{sT} and the output sequence y_1, y_2, …, y_{tT}, and the training is conducted by the guidance of the Hebb learning rules; w_1, w_2, …, w_{uT} are the final obtained link weights. Testing set provides the input sequence x_1, x_2, …, x_{sE}, and the weights w_1, w_2, …, w_{uE} of testing set is selected from w_1, w_2, …, w_{uT} ($uT > uE$); based on the input sequence and judgment criterion, we work out the input y_1, y_2, …, y_{tE} of testing set. After normalizing the link weights w_1, w_2, …, w_{uE} and combing with the output sequence y_1, y_2, …, y_{tE} of testing set, we can dynamically get the total scores of system behavior.

In Fig. 2, by the input sequence and output sequence of the training set, w_1, w_2, …, w_{uT} can be trained, whereas the initial values of w_1, w_2, …, w_{uT} can take 0 and can be also taken by experience. Since the Hebb learning rule is with the effects of gradual optimization, the final w_1, w_2, …, w_{uT} obtained by training is with the better effects. In our work, the initial values of w_1, w_2, …, w_{uT} are computed according to G1 behavior sequence algorithm.

The structure of the data is as follows:

```
Struct Data{
    Struct Training_Set{
        input sequence;
        output sequence;
    };
    Link weights;
    Struct Testing_Set{
        input sequence;
        output sequence;
    };
};
```

Fig. 2 Hebb dynamic behavior assessment model



Example 1 Suppose the length of input sequence of the training set, the length of output sequence of the training set, the length of input sequence of testing set, the length of output sequence of testing set, and the number of link weights are all equal to 4, viz., $sT = tT = sE = tE = uT = uE = 4$. The system has standard behavior in each time, viz., judgment criterion, and the actual system behavior need to be compared by the standard behavior. Known the input sequence of testing set at t_1 is {0.6, 0.5, 0, 0.5}, and the judgment criterion is {{0.5, 0.5, 0.5, 0.5}; [0.01, 0.5]}, and the current value of link weights is {1, 0, 0, 0}. The total score of the actual system behavior at t_1 is require to be work out.

Analysis: in this example, the current values of the input sequence of training set and the link weights are only provided, and the system is needed to be tested and be trained as well.

Firstly, according to formula (5), work out the output sequence of the testing set, viz., $\left\{\frac{0.5-|0.6-0.5|}{0.5-0.01}, 1, 0, 1\right\} = \{0.816, 1, 0, 1\}$.

According to the current values of the link weights and the output sequence of the testing set, the total score of the system at t_1 can be worked out, viz., $1 \times 0.816 \times 100\,\% = 81.6\,\%$.

Secondly, according to formula (4), compute the increase of the link weights, without loss of generality, let $f_i(x_1, x_2, \ldots, x_s) = x_i$, $g_j(y_1, y_2, \ldots, y_t) = y_j$ and $\alpha = 1$, and then the increase of the link weights is $\{0.6 \times 0.816, 0.5 \times 1, 0 \times 0, 0.5 \times 1\} = \{0.49, 0.5, 0, 0.5\}$.

And then, according to formula (4), compute the updated link weights, viz., $\{1 + 0.49, 0 + 0.5, 0, 0 + 0.5\} = \{1.49, 0.5, 0, 0.5\}$.

Finally, conduct the normalization for the updated link weights, viz., {0.598, 0.201, 0, 0.201}, and this weight will be treated as the weight of the total score in the next moment (t_2 moment).

Example 2 Based on Example 1, the input sequence of the testing set at t_2 is {0.8, 0.5, 0, 1}, and the total score of the actual behavior of the system at t_2 is required to be worked out.

Analysis: the link weight at t_2 has been worked out at t_1, and therefore the total score of the system actual behavior can be worked out, directly.

Firstly, according to formula (5), work out the output sequence of the testing set, viz., $\left\{\frac{0.5-|0.8-0.5|}{0.5-0.01}, 1, 0, 0\right\} = \{0.408, 1, 0, 0\}$.

According to the link weight computed at t_1 and the output sequence of the testing set, the total score of the system behavior at t_2 can be worked out, viz., $(0.598 \times 0.408 + 0.201 \times 1 + 0 + 0) \times 100\,\% = 44.5\,\%$.

Secondly, according to formula (4), compute the increase of the link weight, viz., $0.8 \times 0.408, 0.5 \times 1, 0 \times 0, 1 \times 0\} = \{0.326, 0.5, 0, 0\}$.

And then, according to formula (4), compute the updated link weights, viz., $\{0.598 + 0.326, 0.201 + 0.5, 0, 0.201 + 0\} = \{0.924, 0.701, 0, 0.201\}$.

Finally, conduct the normalization for the updated link weights, viz., {0.506, 0.0.384, 0, 0.11}, and the computed weight will be as the weights to compute the total score in the next moment (t_3 moment).

In the same way, since the input sequence of system can be obtained in real time, and the judging criterion for the system is known, we can compute the total scores of the system at the moments t_3, t_4, t_5, ….

If the behavioral assessment is in the distributed network, at least four nodes is needed to be considered: the

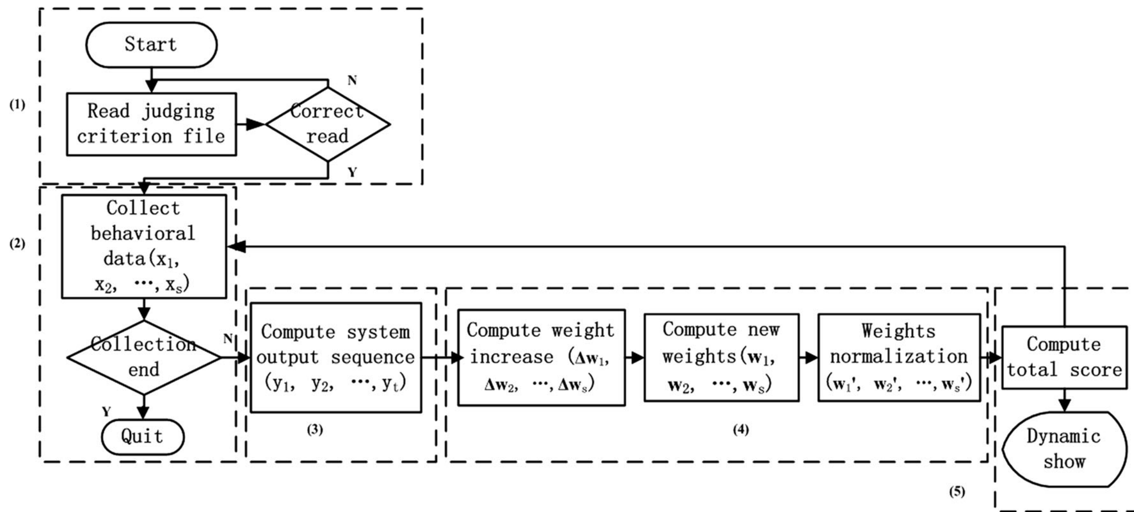


Fig. 3 Dynamic behavior assessment process diagram

first node is treated as the main node to execute the Hebb learning algorithm; the second node is used to solve the kinetic equation of motion and the flight data are transmitted to the main node by TCP/UDP protocol; the third node is used to complete the video rendering and display; training set is put in the fourth node and transmitted to the main node by TCP/UDP protocol.

3.3 Design of algorithms

The process diagram implementing the dynamic behavior assessment model based on Hebb learning rule is shown in Fig. 3.

In Fig. 3, the part (1) denotes reading the system judging criterion file, e.g., the preset air line of the flight control system, the preset regulations of the industrial control system, and so on; the part (2) denotes the dynamic collection for the behavioral data; the part (3) is the computation or obtainment for the system output sequence; the part (4) denotes the link weights recursively computed by Hebb learning rule; and the part (5) denotes computing the total score and dynamic showing.

Based on Fig. 3, we design Algorithms 1, 2 and 3 as follows, where Algorithm 1 is the total algorithm of the system, which calls Algorithms 2 and 3.

Algorithm 1. Dynamic Behavioral Assessment Algorithm

Input: D, system judging criterion file; variables $\{x_1, x_2, \ldots, x_s\}$, behavioral data; $\{r_2, r_3, \cdots, r_s\}$, the relative importance set between objectives

Output: Score, the total assessment score of system behavior

Method:

(1) while(readFileState == true){ // readFileState is the flag that the data are correctly read

(2) x[] ← readFile(D); // read the file of system judging criterion

(3) }

(4) $w_1^0, w_2^0, \ldots, w_s^0$ ← G1($\{x_1, x_2, \cdots, x_s\}$, $\{r_2, r_3, \cdots, r_s\}$);// call Algorithm G1

(5) while(collectState == true){ // if continue collecting

(6) $x_1, x_2, \ldots, x_s$ ← readSensorData(D); // obtain the system behavioral data by readSensorData(D)

(7) for(i = 1 to s){ // s is the number of behavioral data objectives

(8) if(abs(x_i - $x^0{}_i$) > ε_i^+) // compute $y_1, y_2, \ldots, y_s$

(9) y_i = 0;

(10) else if(abs(x_i - $x^0{}_i$)< ε_i^-)

(11) y_i = 100;

(12) else

(13) y_i = $(\varepsilon_i^+ - abs(x_i - x_i^o)) / \varepsilon_i^+ - \varepsilon_i^- \times 100\%$;

(14) }

(15) $w_1, w_2, \ldots, w_s$ ← Hebb($\{x_1, x_2, \cdots, x_s\}$,$\{w_1^0, w_2^0, \ldots, w_s^0\}$, α);// call Algorithm Hebb

(16) Score = 0;

(17) for(j = 1; j < s; j++){

(18) Score = Score + $w_j * y_i$; // compute the total assessment score

(19) }

(20) display Score; // dynamically show Score

(21) $w_1^0, w_2^0, \ldots, w_s^0 \leftarrow w_1, w_2, \ldots, w_s$;

(22) }

Algorithm Explains: in Algorithms, the judging criterion is firstly read in the array $x[]$, and then the G1 behavioral sequence algorithm (Algorithm 2) is called to compute the corresponding initial weights, viz., w_1^0, w_2^0, …, $w_s^0 \leftarrow$ G1($\{x_1, x_2, \ldots, x_s\}$, $\{r_2, r_3, \ldots, r_s\}$); readSensorData(D) is used to obtain the system behavioral data and by dual for-loop, the corresponding evaluation results y_1, y_2, …, y_s are worked out. Finally, by Hebb($\{x_1, x_2, \ldots, x_s\}$,$\{w_1^0, w_2^0, \ldots, w_s^0\}$, α), the updated weights are worked out, where Hebb() is the Hebb learning rule algorithm (Algorithm 3). Steps from (16) to (20) are the display process of the dynamic behavioral assessment.

Algorithm 2. G1 Behavioral Sequence Algorithm

Input: $\{x_1, x_2, \cdots, x_s\}$, assessment objectives set; $\{r_2, r_3, \cdots, r_s\}$, relative importance set between objectives

Output: $\{w_1, w_2, \cdots, w_s\}$, weights set of assessment objectives

Method:

(1) sum ← 0;

(2) for(i = 2 to s){

(3) r ←1; // initiate the temporary variable r

(4) for(j = i to s){ // Multiply the weight ratios

(5) r = r * r_i;

(6) } }

(7) sum = sum + r;

(8) INV_w = 1 + sum; // let INV_w * w_s = 1

(9) w_s = 1 / INV_w;

(10) for(k = s-1 to 2){ // recursively work out $w_{s-1}, w_{s-2}, \ldots, w_1$

(11) $w_k = w_{k+1} * r_{k+1}$;

(12) }

(13) return $\{w_1, w_2, \cdots, w_s\}$;

Algorithm Explains: In Algorithm 2, Eq. (3), viz., $1 + \sum_{i=2}^{s}\prod_{j=i}^{s} r_j$ is computed form steps (1) to (8), where $\prod_{j=i}^{s} r_j$ is computed from steps (4) to (6); according to Eq. (3), w_s is equal to the inverse of the result computed in step (8), hence let $w_s = 1/\mathrm{INV\_}w$. w_{s-1}, w_{s-2}, …, w_1 is recursively worked out from steps (10) to (12); the computed result w_1, w_2, …, w_s is returned in step (13).

Algorithm 3. Hebb Learning Rule Algorithm

Input: $x_1, x_2, \ldots, x_s$, behavioral data of system; $y_1, y_2, \ldots, y_t$, assessment result to system behavioral data;
$w_1^0, w_2^0, \ldots, w_u^0$, input weights; α, learning ratio

Output: $w_1, w_2, \ldots, w_u$, updated weights

Method:

(1) $w_1, w_2, \ldots, w_u \leftarrow w_1^0, w_2^0, \ldots, w_u^0$;

(2) for (i = 1 to u){ // u is the number of input weights

(3) $\Delta w_i = \alpha * f_i(x_1, x_2, \ldots, x_s) * g_i(y_1, y_2, \ldots, y_t)$;

// Δw_i denotes the increase of the i^th^ weight; f_i and g_i are the functions for the taking values of the i^th^ weight

(4) $w_i = w_i + \Delta w_i$;

(5) }

(6) sumW = 0;

(7) for(i = 1 to u){

(8) sumW = w_i + sumW;

(9) }

(10) for (i = 1 to u){

(11) $w_i = w_i$ /sumW;

(12)}

Algorithm Explains: In Algorithm 3, the initial weights are assigned in step (1), and the weights are updated by steps (2)–(5), and the updated weights are normalized from steps (6) to (12).

4 Experiments

In this section, we test and verify the validity of the dynamic behavior assessment model by experiments. The experiment platform is developed in MS VStudio 2010 to assess the flight behavior, which can communicate with the large flight simulation software FlightGear and can collect the behavioral data of pilots in real time.

4.1 Experimental setup

According to the dynamic behavioral assessment model, we select 11 attributes, which include accelerate, angle of attack in taking off. These attributes are used to assess the behavior of pilots and treated as the input sequence x_1, x_2, …, x_{11}, whose corresponding relations are shown in Table 2.

In Table 2, there are totally 11 assessment objectives and their physical meanings are also listed, which will be as the input data of the behavioral assessment system, where x_1 denotes the aircraft acceleration; x_2 denotes the critical angle of attack in taking off; x_3 denotes the speed in taking off; x_4 denotes the altitude of aircraft; x_5 denotes the lift rate of aircraft; x_6 denotes the airspeed of aircraft; x_7 denotes heading angle of aircraft; x_8 denotes the roll angle of aircraft; x_9 denotes the pitch angle of aircraft; x_{10} denotes the landing speed of aircraft; x_{11} denotes the attach angle in landing.

The judging criterion of system is the preset air lines, that is to say, the pilot flies the aircraft according to the preset air lines, and in each moment there are standard x_1^0, x_2^0, …, x_{11}^0 to assess $x_1, x_2, \ldots, x_{11}$. Let the threshold $[\varepsilon^-, \varepsilon^+]$ of the minimal and maximal error is [0.5, 1], according to formula (5) conduct the assessment for x_1, x_2, …, x_{11}, and denote the assessment result to be y_1, y_2, …, y_{11}. For example, the current value of the accelerate x_1 is 10 and its corresponding expected value x_1^0 is 10.8, and then, the assessment result for x_1 is:

$$y_1 = \frac{\varepsilon^+ - (x_1 - x_1^o)}{\varepsilon^+ - \varepsilon^-} \times 100\,\% = \frac{1 - (10.8 - 10)}{1 - 0.5} \times 100\,\% = \frac{0.2}{0.5} \times 100\,\% = 40\,\%$$

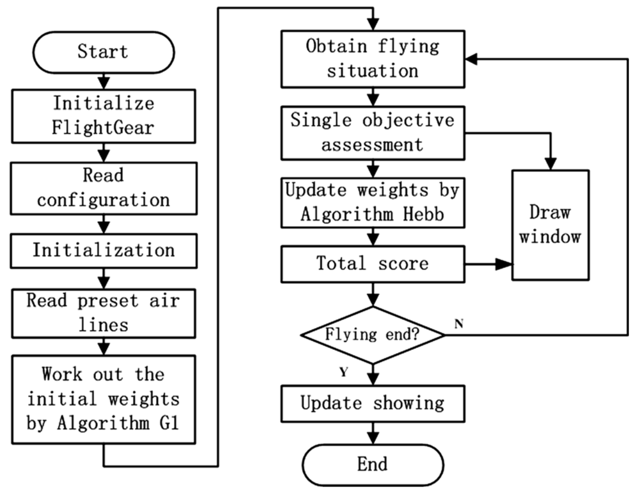


Fig. 4 Flow diagram of dynamic behavioral assessment

According to Hebb learning rule, we can dynamically work out the corresponding link weights w_1, w_2, …, w_{11} of y_1, y_2, …, y_{11}, and the computing method is according to Eq. (4). The initial values of w_1, w_2, …, w_{11} is provided by G1 behavioral sequence algorithm, and in the whole process, the next moment values of w_1, w_2, …, w_{11} can be recursively worked out by the current values of w_1, w_2, …, w_{11}.

We use y_1, y_2, …, y_{11} and w_1, w_2, …, w_{11} to compute the total score, viz., $\sum_{k=1}^{11} w_k y_k$.

The flow diagram is shown in Fig. 4.

In Fig. 4, the simulator with FlightGear is firstly initialized, and the pre-configured log-config files are read and then the geographical environment, type of aircraft, environmental parameters and so on are set, and the initialization is finished. To follow up, open the "flight behavioral dynamic assessment system" and read the preset air lines and the minimal and maximal error threshold. The "flight behavioral dynamic assessment system" calls the G1 behavioral sequence algorithm to compute the initial value of the link weight. In the software FlightGear, the flight simulation starts, and the flight procedure includes five phases: taking off and taxiing, climbing, level flight, landing and approach; FlgightGear software output flight

Table 2 Assessment objectives and physical meaning

Assessment objective	Physical meaning	Assessment objective	Physical meaning
x_1	Accelerate	x_7	Course angle
x_2	Angle of attack in taking off	x_8	Angle of roll
x_3	Speed in taking off	x_9	Angle of pitch
x_4	Altitude	x_{10}	Speed in landing
x_5	Grade	x_{11}	Angle of attack in landing
x_6	Airspeed		

data by interfaces and the "flight behavioral dynamic assessment system" receive data from FlightGear in real time and conduct dynamic scoring and displaying. When the flight is terminated, the system shows the changing curve of the flight achievement.

4.2 Experimental results

We have conducted the dynamic assessment for the behavior of 10 volunteers, and Fig. 5 shows their training curves.

In Fig. 5, it is obviously to show that there are different scores in different phases for pilots, i.e., there are higher scores in the phase of taking off and taxiing; there are medium scores in the phases of climbing and level flight; and there are lower scores in the phases of landing and approach. In addition, theses curves reflect different psychological diathesis and technical qualification of pilots. Based on these curves, we can draw up a training plan in a pointed manner.

We collect the comprehensive scores of each pilot; see also in Table 3.

In Table 3, we list the total scores of 10 pilots obtained from the dynamic behavioral assessment based on Hebb learning rule and the G1 behavioral sequence algorithm, respectively. As shown in the table, the total scores obtained from the dynamic behavioral assessment are higher than the existing method, which means the assessment accuracies of the dynamic behavioral assessment is higher.

4.3 Experimental analysis

Obviously, the dynamic behavior assessment method is with remarkable advantages compared with the behavioral assessment method based on G1 behavioral sequence algorithm, as shown in Table 4.

Table 3 Comprehensive scores of dynamic behavioral assessment and the existing method

Pilots	Total scores based on dynamic behavioral assessment of Hebb learning rule	Total scores based on the existing method (G1 algorithm)
1	73.6418	73.0844
2	88.1000	92.0579
3	91.5697	91.5085
4	80.5425	71.8955
5	78.2827	78.3069
6	96.4167	90.1850
7	86.1408	67.8780
8	92.8308	95.9131
9	85.9937	89.6779
10	80.3375	69.8426

In Table 4, the dynamic behavioral assessment method and the assessment method based on G1 behavioral sequence algorithm are compared from execution efficiency, accuracy, online computing support, and dynamic assessment support, respectively.

1. From the point of view of efficiency, G1 behavioral sequence algorithm need read m times data in each time (*m* is the number of groups collecting the behavioral data), and conduct m times assessment for these m groups of data, whereas the dynamic behavioral assessment model need only read one time data and conduct one time assessment. Therefore, the efficiency of the dynamic behavioral assessment method is higher than G1 behavioral sequence assessment algorithm.
2. From the point of view of accuracy, in the dynamic behavioral assessment model, the weights are obtained by dynamic prediction, and when the phase of flight is switched, the prediction of weights can produce errors

Fig. 5 Training curves of pilots

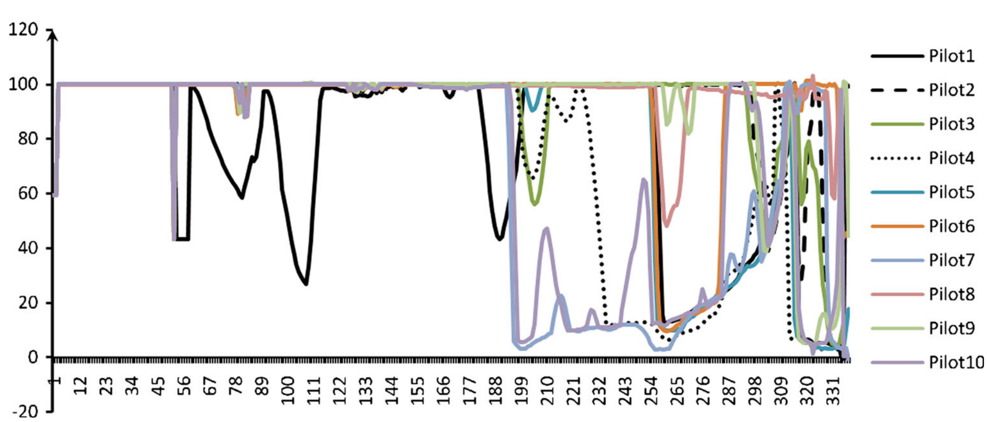


Table 4 Comparisons between dynamic behavioral assessment and G1 assessment

Behavioral assessment	Efficiency	Accuracy	Complexity	Online computing	Dynamic assessment
Dynamic behavioral assessment	High	Higher	Simple	Support	Support
Behavioral assessment based on G1 behavioral sequence algorithm	Low	High	Complex	Not support	Not support

for slower adaptation speed, whereas in G1 behavioral sequence assessment algorithm, the weights are obtained by the computation of G1 algorithm and the revision of experts, hence in the switch of the flight stage, the errors of weights are relatively smaller.

3. From the point of view of algorithm complexity, since G1 behavioral sequence algorithm conduct the assessment in stages, the weights of all objectives in each stage need set in advance, whereas in the dynamic behavioral assessment method, we need only set the initial weight, and then the system can automatically predict the subsequent weights according to Hebb learning rule.
4. From the point of view of online computing support, G1 behavioral sequence algorithm need compute the corresponding weights in stages offline, whereas the dynamic behavioral assessment model can compute the weights of next stage online; therefore, G1 behavioral sequence algorithm does not support online computing, while the dynamic behavioral assessment model supports it.
5. From the point of view of dynamic assessment support, since G1 behavioral sequence algorithm cannot compute the weights online, it cannot support the dynamic assessment, whereas the dynamic behavioral assessment model can compute the weights online, so it supports the dynamic assessment.

5 Conclusion

We have introduced a dynamic behavioral assessment model based on Hebb learning rule, which is an adaptive model synchronizing learning and assessment. This model is with high efficiency, high accuracy, adaptation, dynamics, and other advantages. We firstly introduce and propose a sort of G1 behavioral sequence algorithm suitable for this model, and by G1 behavioral sequence algorithm, we can get the initial weights needed by this model. And then conduct the progressive training for the initial weights by the method of Hebb learning rule. In the course of training, dynamically compute the scores of system behavior; at the end of training, count the scores of the whole process and give suggestions of training. We have built the pilots behavioral assessment system based on FlightGear, and taking pilot behavioral assessment for example, we conducted the experimental testing and verification for the dynamic behavioral assessment model. Experimental results show: the dynamic behavioral assessment model based Hebb learning rule is with remarkable advantages in assessment efficiency, reducing complexity, online computing support, dynamic assessment support, and so force, compared with G1 behavioral sequence method.

Adaptive and intelligent evaluation is a key feature for the dynamic behavioral Assessment based on Hebb learning rule, and we will attack the more adaptive and intelligent evaluation issue.

Acknowledgments The research was supported by Fundamental Research Funds for the Central Universities (Project No. CDJZR14185501), Basic Research on the Frontier and Application of Chongqing City under Grant cstc2015jcyjA40006 and the Jiangsu Key Laboratory of Meteorological Observation and Information Processing (Project No. KDXS1502).

References

1. Xenias D, Axon CJ, Whitmarsh L (2015) UK smart grid development: an expert assessment of the benefits, pitfalls and functions. Renew Energy 81:89–102
2. Teferra K, Shields MD, Hapij A (2014) Mapping model validation metrics to subject matter expert scores for model adequacy assessment. Reliab Eng Syst Saf 132:9–19
3. Nishijima T, Llorens-Martin M, Sanchez Tejeda G et al (2013) Cessation of voluntary wheel running increases anxiety-like behavior and impairs adult hippocampal neurogenesis in mice. Behav Brain Res 245:34–41
4. Stein DS, Munir KM, Karweck AJ (2013) Developmental regression, depression, and psychosocial stress in an adolescent with Down syndrome. J Dev Behav Pediatr 34(3):216–218
5. Kaplanski G, Levy H (2015) Value-at-risk capital requirement regulation, risk taking and asset allocation: a mean–variance analysis. Eur J Finance 21(3):215–241
6. Chen B, Huang SF, Pan G (2015) High dimensional mean–variance optimization through factor analysis. J Multivar Anal 133:140–159
7. Zhang SC, Chen F, Wu XD, Zhang CQ, Wang RL (2012) Mining bridging rules between conceptual clusters. Appl Intell 36(1):108–118
8. Zhang SC (2012) Decision tree classifiers sensitive to heterogeneous costs. J Syst Softw 85(4):771–779
9. Wang XJ, Durugbo C (2013) Analysing network uncertainty for industrial product-service delivery: a hybrid fuzzy approach. Expert Syst Appl 40(11):4621–4636
10. Duque-Ramos A, Tomas Fernandez-Breis J, Iniesta M et al (2013) Evaluation of the OQuaRE framework for ontology quality. Expert Syst Appl 40(7):2696–2703

11. Tansel IN, Demetgul M, Bickraj K et al (2013) Basic computational tools and mechanical hardware for torque-based diagnostic of machining operations. J Intell Manuf 24:147–161
12. Lai ZF, Jiang YF (2007) Learning action model in AI planning based on genetic algorithm. Chin J Comput 30(6):945–953
13. Arroyo JM, Fernandez FJ (2013) Application of a genetic algorithm to $n - K$ power system security assessment. Int J Electr Power Energy Syst 49(1):114–121
14. Luque-Baena RM, Elizondo D, Lopez-Rubio E, Palomo EJ, Watson T (2013) Assessment of geometric features for individual identification and verification in biometric hand systems. Expert Syst Appl 40(9):3580–3594
15. Wang T, Qin ZX, Zhang SC, Zhang CQ (2012) Cost-sensitive classification with inadequate labeled data. Inf Syst 37:508–516
16. Knight C, Lindberg GE, Voth GA (2012) Multiscale reactive molecular dynamics. J Chem Phys 137:1–15
17. Huang H, Xiao JC, Yang QS (2013) Creating process-agents incrementally by mining process asset library. Inf Sci 233(1):183–199
18. Zhou L, Chen XP, Huang G, Sun YC, Mei H (2008) Negotiation-enabled modeling and verification of architectural behavior of internetware. Chin J Softw 15(5):1099–1112
19. Hebb D (1949) Organization of behavior. Wiley, New York
20. Yin Y, Gong G, Han L (2011) Theory and techniques of data mining in cgf behavior modeling. Sci China Inf Sci 54(4): 717–731
21. Yin Y, Gong G, Han L (2009) Control approach to rough set reduction. Comput Math Appl 57(1):117–126
22. Gao WQ, Zhang FC, Wang LB, Liu H (2011) Flight training performance assessment model and implementation. Electr Des Eng 19(24):50–52
23. Nunes TM, de Albuquerque VHC, Papa JP (2013) Automatic microstructural characterization and classification using artificial intelligence techniques on ultrasound signals. Expert Syst Appl 40(8):3096–3105