Intuitionistic Fuzzy Petri Nets Model Based on Back Propagation Algorithm for Information Services

Junhua Xi[1, *], Kouquan Zheng[1], Jianfeng Ma[1], Jungang Yang[1] and Zhiyao Liang[2]

Abstract: Intuitionistic fuzzy Petri net is an important class of Petri nets, which can be used to model the knowledge base system based on intuitionistic fuzzy production rules. In order to solve the problem of poor self-learning ability of intuitionistic fuzzy systems, a new Petri net modeling method is proposed by introducing BP (Error Back Propagation) algorithm in neural networks. By judging whether the transition is ignited by continuous function, the intuitionistic fuzziness of classical BP algorithm is extended to the parameter learning and training, which makes Petri network have stronger generalization ability and adaptive function, and the reasoning result is more accurate and credible, which is useful for information services. Finally, a typical example is given to verify the effectiveness and superiority of the parameter optimization method.

Keywords: Intuitionistic fuzzy set, intuitionistic fuzzy Petri nets, production rule, BP algorithm.

1 Introduction

In the objective world, many knowledges difficult to describe in an accurate way. Intuitionistic fuzzy production rules use intuitionistic fuzzy conditional sentences of IF-THEN structure to express uncertain knowledge, which overcomes the defect of single membership degree of fuzzy theory. Its mathematical description is more in line with the ambiguous nature of objective reality, and provides new ideas and methods for the description and processing of uncertain information. Intuitionistic fuzzy Petri net is a good modeling tool. It is based on intuitionistic fuzzy production rule knowledge base system. It can integrate the expression of uncertain knowledge with intuitionistic fuzzy reasoning. It not only improves the reliability of the system, but also makes the knowledge expression, analysis, testing and decision support more convenient. However, the poor self-learning and adjustment ability is the inherent shortcoming of the fuzzy system. Some parameters of intuitionistic fuzzy production rules, such as weights, thresholds, credibility, are difficult to obtain accurately or even cannot be obtained, which reduces the knowledge reasoning and generalization ability of IFPN (Intuitionistic Fuzzy Petri Net). Neural network has strong self-adaptive and self-learning ability. Its BP

[1] College of Information and Communication, National University of Defense Technology, Xi'an, 710106, China.

[2] Macau University of Science and Technology, Taipa, Macau.

[*] Corresponding Author: Junhua Xi. Email: 1011009123@163.com.

Received: 10 February 2019; Accepted: 26 July 2019.

algorithm can approximate any non-linear mapping relationship and converge to the extreme point of corresponding surface quickly. It is a common preferred optimization strategy. How to integrate the learning function of the neural network into IFPN, through the study and training of a batch of sample data, make the parameters get rid of the dependence on experience, more in line with the actual system, which has become an important problem to be solved urgently.

Most of the studies on learning ability of fuzzy Petri nets at home and abroad are based on strict restrictions or application scope. Although the time is not long, a lot of research results have been achieved. Tsang [Tsang (1999)] studied the self-learning ability of FPN by combining the neural network algorithm. By modifying the traditional BP algorithm to realize forward reasoning and reverse weight adjustment, a self-learning algorithm of weight and confidence without threshold was proposed. Li et al. [Li, Yu and Felipe (2000); Li and Felipe (2000)] proposes an adaptive FPN model, which uses neural networks to perform self-learning operations of weights. Literature Wu et al. [Wu, Er and Gao (2001)] expands the scope of application of "or" rule of FPN model, so that each transition has independent confidence and threshold, and introduce ant colony algorithm and genetic algorithm into the parameter optimization of FPN, so that the accuracy of parameter optimization is higher. Literature Sun et al. [Sun and Wang (2018); Liu, You, Li et al. (2017)] study intuitionistic fuzziness based on Petri nets and applies it to knowledge representation and reasoning

IFPN combines the advantages of intuitionistic fuzzy sets and FPN theory, enhances the model's ability to deal with practical problems, and greatly improves the reasoning accuracy. Because of the increase of membership constraint, the optimization of its parameters is more complicated. Simply referring to the optimization scheme of FPN model will increase the complexity of the system and make the algorithm converge slowly and easily fall into local optimal solution. In view of this, this paper extends the intuitionistic fuzzy of BP algorithm in the neural network to the IFPN model parameter optimization process, and proposes a BP-IFPN parameter optimization method. Through repeated learning and training of the system parameters, the precision of parameter optimization is improved, and the model has stronger generalization ability. Finally, an example is given to illustrate the effectiveness of the method.

2 Intuitionistic fuzzy Petri nets

2.1 Intuitionistic fuzzy Petri nets

Definition 1 (IFPN model) The model structure of intuitionistic fuzzy Petri nets can be represented by the following six tuples:

$$IFPN=(P, T, I, O, \tau, \theta) \tag{1}$$

Among them, $P=\{p_1, p_2, \cdots, p_n\}$ is a collection of finite libraries, each library represents a proposition; $T=\{t_1, t_2, \cdots, t_m\}$ is a collection of limited transitions, each transition represents a rule; $I : P\times T \to [0,1]$ is a $n\times m$ dimensional weighted input matrix, whose element a_{ij} satisfies: if p_i is the input of T_j, then $a_{ij}=\omega_{ij}$, otherwise 0, where ω_{ij} represents the weight of the influence of input library p_i on transition T_j ,and $\forall j \in \{1, 2, \cdots, m\}, \sum_i \omega_{ij} = 1$; $O : P\times T \to [0,1]$ is the $n\times m$ output matrix, whose element

b_{ij} satisfies: if p_i is the output of T_j, then $b_{ij}=C_j$, otherwise <0,1>, among them, $C_j=<C\mu_j,C\gamma_j>$ represents the credibility of transition T_j, $C\mu_j$ represents the degree of confidence (confidence level), $C\gamma_j$ represents the opposite degree of confidence (non-confidence level); $\tau=\{\tau_1, \tau_2, \cdots, \tau_m\}$ represents the transition threshold, that is, the condition for starting the inference rule, where $\tau_j=<\alpha_j,\beta_j>$, and $0<\alpha_j+\beta_j\leq 1, 0<\alpha_j\leq 1, 0<\beta_j\leq 1$, α_j is a threshold of confidence level, β_j is a threshold of non-confidence level; $\theta: P \to [0,1]$ is an association function of the library P, representing the fuzzy truth value of a proposition, the initial Token value is $\theta^0 = \{\theta_1^0, \theta_2^0, \cdots, \theta_n^0\}$,where $\theta_i^0 =< \theta\mu_i^0, \theta\gamma_i^0 >$.

Intuitionistic fuzzy production rules are used to describe intuitionistic fuzzy relations among propositions, literature Shen et al. [Shen, Lei and Li (2009)] gives two IFPN models corresponding to production rule of "merge" and "analysis", which based on intuitionistic fuzzy. The activation of intuitionistic fuzzy production rules is realized through the ignition of transition. For arbitrary transitions, transitions are enabled if the sum of the mark value of input library and the weights on the corresponding input arcs is greater than or equal to the threshold of transitions. We transform the problem into a continuous function whose independent variables satisfy certain requirements, so that the result of intuitionistic fuzzy reasoning can be a continuous function which is convenient for first-order derivation. Referring to the Sigmoid function in literature [Li, Yu and Felipe (2000)], we establish transitions to ignite intuitionistic fuzzy reasoning continuous functions.

2.2 Intuitionistic fuzzy reasoning function

Let $f(x,y)$ be a function of two variables, b is a constant, and its expression is:

$$f(x, y) = [(1+e^{-b(x-\tau_1)})(1+e^{b(y-\tau_2)})]^{-1} \tag{2}$$

When b is large enough, if $x > \tau_1$ and $y < \tau_2$, $e^{-b(x-\tau_1)} \approx 0$, $e^{b(y-\tau_2)} \approx 0$,then $f(x, y) \approx 1$; if $x < \tau_1$ or $y > \tau_2$, $e^{-b(x-\tau_1)} \to \infty$ or $e^{b(y-\tau_2)} \to \infty$, then $f(x, y) \approx 0$. Obviously, this continuous function $f(x, y)$ can be used as a sign to judge whether change is feasible.

2.2.1 Transition ignites continuous function

Referring to the IFPN model corresponding to production rule based on intuitionistic fuzzy combination in literature [Shen, Lei and Li (2009)], let $x = x_\mu = \sum_{j=1}^{n} M_\mu(p_{ij}) \times \omega_{ij}$, $y = x_\gamma = \sum_{j=1}^{n} M_\gamma(p_{ij}) \times \omega_{ij}$, $\tau_1 = \tau\mu_i$, $\tau_2 = \tau\gamma_i$,then the function $f(x_\mu, x_\gamma) = [(1+e^{-b(x_\mu-\tau\mu_i)})(1+e^{b(x_\gamma-\tau\gamma_i)})]^{-1}$ establishes the judgment of transition enabled. When b is large enough, from the above analysis, we can see: if $x_\mu > \tau\mu_i$ and $x_\gamma < \tau\gamma_i$, $f(x, y) \approx 1$,a transition t_i can be enabled; if $x_\mu < \tau\mu_i$ or $x_\gamma > \tau\gamma_i$, $f(x, y) \approx 0$, a transition t_i cannot be ignited. So, we can use continuous function

$f(x_\mu, x_\gamma)C\sum_{j=1}^{n}(M_\mu(p_{ij})\times\omega_{ij})$, indicate whether transition t_i is ignited, and the Token value of p in its output library can be defined as:

$$\begin{cases} M_\mu(p) = G_\mu(x_\mu, x_\gamma) = f(x_\mu, x_\gamma)C_i\sum_{j=1}^{n}(M_\mu(p_{ij})\times\omega_{ij}) \\ M_\gamma(p) = G_\gamma(x_\mu, x_\gamma) = 1 - f(x_\mu, x_\gamma)C_i\sum_{j=1}^{n}(M_\mu(p_{ij})\times\omega_{ij}) - M_\pi(p) \end{cases} \tag{3}$$

Among them, $M_\mu(p), M_\gamma(p), M_\pi(p)$ is the membership degree, non-membership degree and hesitation value of p respectively, $M_\mu(p), M_\gamma(p), M_\pi(p)\in[0,1]$ and $M_\mu(p)+M_\gamma(p)+M_\pi(p)=1$; $C_i=(1+C\mu_i-C\gamma_i)/2$. For the rule of intuitionistic fuzzy analysis, the following maximum / minimum operation continuous functions can also be established.

2.2.2 Maximum / minimum operation continuous function

Let $y(x)=1/(1+e^{-b(x-k)})$, x_1, x_2, x_3, x_4 be the output value of the transition enabled time, when b is large enough, obviously, the following inference is correct:

$t=\max(x_1,x_2)\approx x_1/(1+e^{-b(x_1-x_2)})+x_2/(1+e^{-b(x_2-x_1)})$;

$h=\max(x_1,x_2,x_3)=\max(t,x_3)\approx t/(1+e^{-b(t-x_3)})+x_3/(1+e^{-b(x_3-t)})$;

$g=\max(x_1,x_2,x_3,x_4)=\max(h,x_4)\approx h/(1+e^{-b(h-x_4)})+x_4/(1+e^{-b(x_4-h)})$;

In the same argument, let $\Phi(x)=1/(1+e^{b(x-k)})$, when b is large enough, obviously, the following inference is correct:

$t=\min(x_1,x_2)\approx x_1/(1+e^{b(x_1-x_2)})+x_2/(1+e^{b(x_2-x_1)})$;

$h=\min(x_1,x_2,x_3)=\min(t,x_3)\approx t/(1+e^{b(t-x_3)})+x_3/(1+e^{b(x_3-t)})$;

$g=\min(x_1,x_2,x_3,x_4)=\min(h,x_4)\approx h/(1+e^{b(h-x_4)})+x_4/(1+e^{b(x_4-h)})$.

After each transition, the Token value of p in the IFPN output library is:

$$\begin{cases} M_\mu(p) = G_\mu(x_\mu, x_\gamma) = f(x_\mu, x_\gamma)\cdot C_i\cdot M_\mu(p_{ij}) \\ M_\gamma(p) = G_\gamma(x_\mu, x_\gamma) = 1 - f(x_\mu, x_\gamma)\cdot C_i\cdot M_\mu(p_{ij}) - M_\pi(p) \end{cases} \tag{4}$$

By analogy, we use maximum and minimum arithmetic continuous functions. When there is several transition enablement, the corresponding output library p can always get a continuous function to obtain the maximum membership and minimum non-membership values. After establishing the intuitionistic fuzzy reasoning function, we can learn and modify the parameters of IFPN.

3 Intuitionistic fuzzy Petri nets based on BP algorithm

The IFPN model can be regarded as composed of nodes, each node centered on transition. Following the BP algorithm in the neural network, the BP algorithm is used to design and adjust parameters for each IFPN node at each level. The IFPN model is divided into l layers, n libraries, and d transitions, with b termination libraries p_k (k=1,2,⋯,b)， r batch sample data is used to learn, and each batch of input samples is s, and the corresponding error function is:

$$E_s = [\sum_{k=1}^{b}((M_\mu(p_k)^s - M_\mu^*(p_k)^s)^2 + (M_\gamma(p_k)^s - M_\gamma^*(p_k)^s)^2)]/2 \tag{5}$$

Among them, $< M_\mu(p_k)^s, M_\gamma(p_k)^s >, < M_\mu^*(p_k)^s, M_\gamma^*(p_k)^s >$ are the actual and expected outputs of the membership and non-membership degrees of the termination repository p_k under the action of sample s. The average error function of r batch samples is:

$$E = \frac{1}{2r}\sum_{s=1}^{r}[\sum_{k=1}^{b}((M_\mu(p_k)^s - M_\mu^*(p_k)^s)^2 + (M_\gamma(p_k)^s - M_\gamma^*(p_k)^s)^2)] \tag{6}$$

3.1 Intuitionistic fuzzy extension of BP algorithm

The parameters should be adjusted according to the gradient of E_s function in order to converge the network. According to formula (5), according to the steepest descent method, the correction formula of the parameters of layer l can be obtained as follows: according to formula (5), the formula of the parameters of layer l can be obtained by the steepest descent method:

$$\Delta w_{ij}^l = -\eta\frac{\partial E_s}{\partial w_{ij}^l} = -\eta(\frac{\partial E_s}{\partial M_\mu^l(p_k)^s}\frac{\partial M_\mu^l(p_k)^s}{\partial w_{ij}^l} + \frac{\partial E_s}{\partial M_\gamma^l(p_k)^s}\frac{\partial M_\gamma^l(p_k)^s}{\partial w_{ij}^l}) \tag{7}$$

$$\begin{cases} \Delta C\mu_i^l = -\eta\frac{\partial E_s}{\partial C\mu_i^l} = -\eta(\frac{\partial E_s}{\partial M_\mu^l(p_k)^s}\frac{\partial M_\mu^l(p_k)^s}{\partial C\mu_i^l} + \frac{\partial E_s}{\partial M_\gamma^l(p_k)^s}\frac{\partial M_\gamma^l(p_k)^s}{\partial C\mu_i^l}) \\ \Delta C\gamma_i^l = -\eta\frac{\partial E_s}{\partial C\gamma_i^l} = -\eta(\frac{\partial E_s}{\partial M_\mu^l(p_k)^s}\frac{\partial M_\mu^l(p_k)^s}{\partial C\gamma_i^l} + \frac{\partial E_s}{\partial M_\gamma^l(p_k)^s}\frac{\partial M_\gamma^l(p_k)^s}{\partial C\gamma_i^l}) \end{cases} \tag{8}$$

$$\begin{cases} \Delta \tau\mu_i^l = -\eta\frac{\partial E_s}{\partial \tau\mu_i^l} = -\eta(\frac{\partial E_s}{\partial M_\mu^l(p_k)^s}\frac{\partial M_\mu^l(p_k)^s}{\partial \tau\mu_i^l} + \frac{\partial E_s}{\partial M_\gamma^l(p_k)^s}\frac{\partial M_\gamma^l(p_k)^s}{\partial \tau\mu_i^l}) \\ \Delta \tau\gamma_i^l = -\eta\frac{\partial E_s}{\partial \tau\gamma_i^l} = -\eta(\frac{\partial E_s}{\partial M_\mu^l(p_k)^s}\frac{\partial M_\mu^l(p_k)^s}{\partial \tau\gamma_i^l} + \frac{\partial E_s}{\partial M_\gamma^l(p_k)^s}\frac{\partial M_\gamma^l(p_k)^s}{\partial \tau\gamma_i^l}) \end{cases} \tag{9}$$

Among them, $\partial E_s / \partial w_{ij}^l$ is E_s 's growth rate of w_{ij}^l. In order to reduce the error, Δw_{ij}^l is inversely proportional to it. If $\partial E_s / \partial w_{ij}^l > 0$, the current position of the system is on the right side of the minimum point, the value of w_{ij}^l should be reduced; otherwise, the position of the system on the left side of the minimum point, the value of w_{ij}^l should be increased; $<\partial E_s/\partial C\mu_i^l, \partial E_s/\partial C\gamma_i^l>$ is the growth rate of E_s for confidence and non-confidence; $<\partial E_s/\partial \tau\mu_i^l, \partial E_s/\partial \tau\gamma_i^l>$ is the growth rate of E_s for confidence and non-confidence; $\eta \in (0,1)$ is the learning coefficient. Let $t_i^l \in T_l$ (i=1, 2, ⋯, d) is a transition of the 1st layer of IFPN. The weighting coefficients on the input arc of t_i^l are $w_{i1}^l, w_{i2}^l, \cdots, w_{im}^l$ respectively. The threshold is $\tau_i^l = <\tau\mu_i^l, \tau\gamma_i^l>$ and the reliability is $C_i^l = <C\mu_i^l, C\gamma_i^l>$. If $p_k^l \in O(t_i^l)$, p_k^l is to terminate the library, then we can calculate the gradient of the parameters:

1) weight coefficient:

$$\begin{aligned}
\frac{\partial E_s}{\partial w_{ij}^l} &= \frac{\partial E_s}{\partial M_\mu^l(p_k)^s}\frac{\partial M_\mu^l(p_k)^s}{\partial w_{ij}^l} + \frac{\partial E_s}{\partial M_\gamma^l(p_k)^s}\frac{\partial M_\gamma^l(p_k)^s}{\partial w_{ij}^l} \\
&= (M_\mu(p_k)^s - M_\mu^*(p_k)^s)\frac{\partial M_\mu^l(p_k)^s}{\partial w_{ij}^l} + (M_\gamma(p_k)^s - M_\gamma^*(p_k)^s)\frac{\partial M_\gamma^l(p_k)^s}{\partial w_{ij}^l} \\
&= \frac{(M_\mu(p_k)^s - M_\mu^*(p_k)^s)C_i}{(1+e^{-b(x_\mu(t_i)-\tau\mu_i)})(1+e^{b(x_\gamma(t_i)-\tau\gamma_i)})}\left(M_\mu(p_x) + b\frac{M_\mu(p_x)e^{-b(x_\mu(t_i)-\tau\mu_i)}}{(1+e^{-b(x_\mu(t_i)-\tau\mu_i)})} - b\frac{M_\gamma(p_x)e^{b(x_\gamma(t_i)-\tau\gamma_i)}}{(1+e^{b(x_\gamma(t_i)-\tau\gamma_i)})}\right) \\
&\quad - \frac{(M_\gamma(p_k)^s - M_\gamma^*(p_k)^s)C_i}{(1+e^{-b(x_\mu(t_i)-\tau\mu_i)})(1+e^{b(x_\gamma(t_i)-\tau\gamma_i)})}\left(M_\mu(p_x) + b\frac{M_\mu(p_x)e^{-b(x_\mu(t_i)-\tau\mu_i)}}{(1+e^{-b(x_\mu(t_i)-\tau\mu_i)})} - b\frac{M_\gamma(p_x)e^{b(x_\gamma(t_i)-\tau\gamma_i)}}{(1+e^{b(x_\gamma(t_i)-\tau\gamma_i)})}\right)
\end{aligned}$$

2) credibility:

$$\begin{cases}
\dfrac{\partial E_s}{\partial C\mu_i^l} = \dfrac{(M_\mu(p_k)^s - M_\mu^*(p_k)^s)x_\mu(t_i)}{2(1+e^{-b(x_\mu(t_i)-\tau\mu_i)})(1+e^{b(x_\gamma(t_i)-\tau\gamma_i)})} - \dfrac{(M_\gamma(p_k)^s - M_\gamma^*(p_k)^s)x_\mu(t_i)}{2(1+e^{-b(x_\mu(t_i)-\tau\mu_i)})(1+e^{b(x_\gamma(t_i)-\tau\gamma_i)})} \\
\dfrac{\partial E_s}{\partial C\gamma_i^l} = -\dfrac{(M_\mu(p_k)^s - M_\mu^*(p_k)^s)x_\mu(t_i)}{2(1+e^{-b(x_\mu(t_i)-\tau\mu_i)})(1+e^{b(x_\gamma(t_i)-\tau\gamma_i)})} + \dfrac{(M_\gamma(p_k)^s - M_\gamma^*(p_k)^s)x_\mu(t_i)}{2(1+e^{-b(x_\mu(t_i)-\tau\mu_i)})(1+e^{b(x_\gamma(t_i)-\tau\gamma_i)})}
\end{cases}$$

3) threshold:

$$\begin{cases}
\dfrac{\partial E_s}{\partial \tau\mu_i^l} = \dfrac{-bC_i x_\mu(t_i)e^{-b(x_\mu(t_i)-\tau\mu_i)}[(M_\mu(p_k)^s - M_\mu^*(p_k)^s) - (M_\gamma(p_k)^s - M_\gamma^*(p_k)^s)]}{(1+e^{-b(x_\mu(t_i)-\tau\mu_i)})^2(1+e^{b(x_\gamma(t_i)-\tau\gamma_i)})} \\
\dfrac{\partial E_s}{\partial \tau\gamma_i^l} = \dfrac{bC_i x_\mu(t_i)e^{b(x_\gamma(t_i)-\tau\gamma_i)}[(M_\mu(p_k)^s - M_\mu^*(p_k)^s) - (M_\gamma(p_k)^s - M_\gamma^*(p_k)^s)]}{(1+e^{-b(x_\mu(t_i)-\tau\mu_i)})(1+e^{b(x_\gamma(t_i)-\tau\gamma_i)})^2}
\end{cases}$$

Let $t_i^{l-1} \in T_{l-1}$ is a transition of the l-1 layer of IFPN, $p_x^{l-1} \in O(t_i^{l-1})$. The weighting coefficients on the input arc of t_i^l are $w_{i1}^l, w_{i2}^l, \cdots, w_{im}^l$ respectively. The threshold is $\tau_i^l = <\tau\mu_i^l, \tau\gamma_i^l>$ and the reliability is $C_i^l = <C\mu_i^l, C\gamma_i^l>$. If $p_k^l \in O(t_i^l)$, p_k^l is to terminate the library, then we can calculate the gradient of the parameters: If p_x^{l-1} is to terminate the library, it can be calculated by reference to the above method; otherwise, $\exists t_i^l \in T_l,\ p_x^{l-1} \in I(t_i^l),\ p_k^l \in O(t_i^l)$, p_k^l will terminate the repository. Let $\delta_\mu^{(l)} = \partial E_s / \partial M_\mu^l(p_k)^s, \delta_\gamma^{(l)} = \partial E_s / \partial M_\gamma^l(p_k)^s$, the gradient of the parameters on the L-1 layer is:

1) weight coefficient:

$$\partial E_s / \partial w_{ij}^{l-1} = \delta_\mu^{(l-1)}[\partial M_\mu^{l-1}(p_x)^s / \partial w_{ij}^l] + \delta_\gamma^{(l-1)}[\partial M_\gamma^{l-1}(p_x)^s / \partial w_{ij}^{l-1}]$$

2) credibility:

$$\begin{cases} \partial E_s / \partial C\mu_i^{l-1} = \delta_\mu^{(l-1)}[\partial M_\mu^{l-1}(p_x)^s / \partial C\mu_i^{l-1}] + \delta_\gamma^{(l-1)}[\partial M_\gamma^{l-1}(p_x)^s / \partial C\mu_i^{l-1}] \\ \partial E_s / \partial C\gamma_i^{l-1} = \delta_\mu^{(l-1)}[\partial M_\mu^{l-1}(p_x)^s / \partial C\gamma_i^{l-1}] + \delta_\gamma^{(l-1)}[\partial M_\gamma^{l-1}(p_x)^s / \partial C\gamma_i^{l-1}] \end{cases}$$

3) threshold:

$$\begin{cases} \partial E_s / \partial \tau\mu_i^{l-1} = \delta_\mu^{(l-1)}[\partial M_\mu^{l-1}(p_x)^s / \partial \tau\mu_i^{l-1}] + \delta_\gamma^{(l-1)}[\partial M_\gamma^{l-1}(p_x)^s / \partial \tau\mu_i^{l-1}] \\ \partial E_s / \partial \tau\gamma_i^{l-1} = \delta_\mu^{(l-1)}[\partial M_\mu^{l-1}(p_x)^s / \partial \tau\gamma_i^{l-1}] + \delta_\gamma^{(l-1)}[\partial M_\gamma^{l-1}(p_x)^s / \partial \tau\gamma_i^{l-1}] \end{cases}$$

Among them, $\delta_\mu^{(l-1)} = \delta_\mu^{(l)}[\partial M_\mu^l(p_k)^s / \partial M_\mu^{l-1}(p_x)^s], \delta_\gamma^{(l-1)} = \delta_\gamma^{(l)}[\partial M_\gamma^l(p_k)^s / \partial M_\gamma^{l-1}(p_x)^s]$, there is a continuous function of $p_x \rightarrow p_k$ in the process of model reasoning, so $\partial M_\mu^l(p_k)^s / \partial M_\mu^{l-1}(p_x)^s$ has solutions. Accordingly, the $h = l-2, \cdots, 1$ layers are recursively calculated in turn, and $\partial E_s / \partial w_{ij}^h, \partial E_s / \partial C\mu_i^h, \partial E_s / \partial C\gamma_i^h, \partial E_s / \partial \tau\mu_i^h$ and $\partial E_s / \partial \tau\gamma_i^h$ are calculated. After obtaining the required gradient, the parameter adjustment learning process of each transition is given:

1) weight coefficient learning process: $w_{ij}^h(q+1) = w_{ij}^h(q) - \eta(\partial E_s / \partial w_{ij}^h)$

2) the credibility learning process: $\begin{cases} C\mu_i^h(q+1) = C\mu_i^h(q) - \eta(\partial E_s / \partial C\mu_i^h) \\ C\gamma_i^h(q+1) = C\gamma_i^h(q) - \eta(\partial E_s / \partial C\gamma_i^h) \end{cases}$

3) threshold learning process: $\begin{cases} \tau\mu_i^h(q+1) = \tau\mu_i^h(q) - \eta(\partial E_s / \partial \tau\mu_i^h) \\ \tau\gamma_i^h(q+1) = \tau\gamma_i^h(q) - \eta(\partial E_s / \partial \tau\gamma_i^h) \end{cases}$

Among them, q denotes the number of times of learning, $w_{ij}^h(q)$ denotes the weight of input place p_x corresponding to transition t_i in the h layer after q learning, and $\sum_{j=1}^{m} w_{ij}^h(q+1) = 1$; $\eta \in (0,1)$ is the learning coefficient.

3.2 Learning rate adjustment

The main disadvantage of applying BP algorithm on IFPN model is slow convergence. Therefore, in order to enhance the robustness of the learning process, reduce the impact of large errors of individual disturbed points and improve convergence, the learning rate is adjusted to:

$$\eta(q)=\eta_0 / [1-SE_s(q)] \tag{10}$$

Among them, η_0 is the initial learning rate; $SE_s(q)$ indicates the error of terminating the library after q learning. In the process of parameter learning, the choice of learning rate η is very important. The larger η is, the faster convergence speed is, but it may cause oscillation due to instability. η small can avoid instability, but the rate of convergence is slow. The simplest way to solve this problem is to add a "momentum term", that is, to add an item in the learning process of each parameter. Its revised formula is as follows:

$$w_{ij}^h(q+1) = w_{ij}^h(q) - \eta(\partial E_s / \partial w_{ij}^h) + \xi w_{ij}^h(q) \tag{11}$$

$$\begin{cases} C\mu_i^h(q+1) = C\mu_i^h(q) - \eta(\partial E_s / \partial C\mu_i^h) + \xi C\mu_i^h(q) \\ C\gamma_i^h(q+1) = C\gamma_i^h(q) - \eta(\partial E_s / \partial C\gamma_i^h) + \xi C\gamma_i^h(q) \end{cases} \tag{12}$$

$$\begin{cases} \tau\mu_i^h(q+1) = \tau\mu_i^h(q) - \eta(\partial E_s / \partial \tau\mu_i^h) + \xi\tau\mu_i^h(q) \\ \tau\gamma_i^h(q+1) = \tau\gamma_i^h(q) - \eta(\partial E_s / \partial \tau\gamma_i^h) + \xi\tau\gamma_i^h(q) \end{cases} \tag{13}$$

Among them, $\xi \in [0,1)$ is a regulatory factor. In order to avoid oscillation and accelerate convergence speed, if $SE_s(q) \le 0$, take $\xi = 0$.

3.3 Learning and training algorithm of Petri net model

According to the extended BP algorithm and IFPN reasoning process, the learning and training steps of IFPN model are given. The learning algorithm proposed in this paper is applicable to the loop-free IFPN model. It needs to be hierarchical. Firstly, the hierarchical algorithm of IFPN model is given.

Algorithm
Input: The initial Token value of the library $\theta^{(0)}$, the initial values and thresholds of learning parameters, such as weight coefficient, transition threshold and reliability, ε;
Output: Terminate the value θ of the library Token, the learning value of each parameter, and the learning times q.
1: Give initial values to parameters that need to be learned, Preprocessing r batch sample data, $q=0$;
2: For r batch sample data, Token values of each library are calculated;
3: Calculating the mean error function E of r batch samples;
4: While ($E > \varepsilon$)
5: Begin
6: Using the output of algorithm 1, the parameters are adjusted, $q=q+1$;
7: For r batch sample data, For library P, For transition T, For r batch sample data, for repository P, transition T, given set $H=\{p_k\}$, p_k is the initial library,

$\forall p_k \in P-H$, let $<M_\mu(p_k), M_\gamma(p_k)> = <0,1>$, i is the number of layers of the model, initially 1;

8: Establishing Change Set $T_i = \{t_j \in T \mid \forall p_k \in I(t_j), p_k \in H\}$;

9: if $t_j \in T_i, t \in T - T_i, \exists p \in O(t_j) \cap O(t)$, then $T_i = T_i - \{t_j\}$;

10: $H = H \cup \{p \mid p \in O(t_j), t_j \in T_i\}$, the output libraries of all changes in T_i are inserted into set H;

11: Using formula (3) (4) to ignite all transitions in T_i layer by layer.

12: Calculating the mean error function E of r batch samples

13: End

14: Obtain the adjusted values of each parameter, the termination repository, the total error function value and the number of times of learning, and then terminate the algorithm.

The time complexity of the algorithm is related to the number of samples, the number of learning times, the number of model layers and the number of changes in each layer. The model layers are known, and the analysis shows that the time complexity of the algorithm is $O(n^3)$.

4 Evaluations

In order to verify the feasibility of the algorithm, the learning and training steps of IFPN are given with examples in literature [Li and Le (2007)], and the reasoning results are compared and analyzed. Let the known libraries p_1, p_2, p_3, p_4, p_5, p_6, p_7 and p_8 correspond to a proposition in the expert system respectively. There are the following fuzzy production rules between them:

R_1: IF p_1 Then p_2 (C_2, τ_2);

R_2: IF p_1 or p_2 Then p_3 $(C_1, \tau_1, C_3, \tau_3)$;

R_3: IF p_3 and p_4 and p_5 Then p_6 $(\omega_{41}, \omega_{42}, \omega_{43}, C_4, \tau_4)$;

R_4: IF p_6 and p_7 Then p_8 $(\omega_{51}, \omega_{52}, C_5, \tau_5)$.

According to the IF production rule and the knowledge representation method of IFPN model, the IFPN model as shown in Fig. 1 can be established.

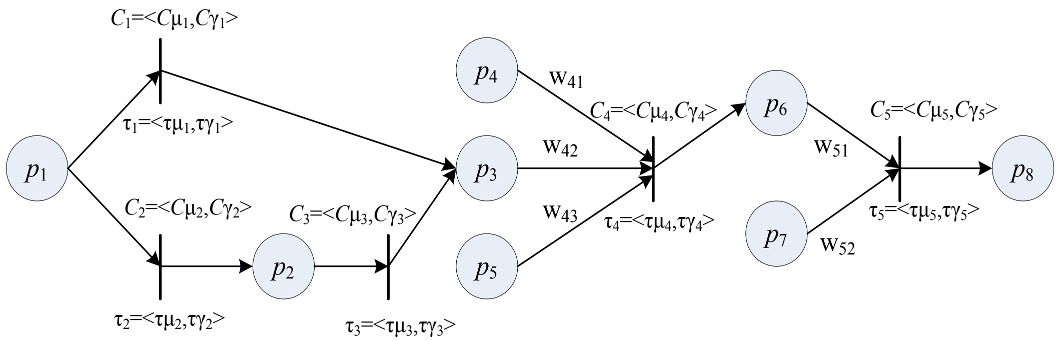


Figure 1: Intuitionistic fuzzy Petri net model

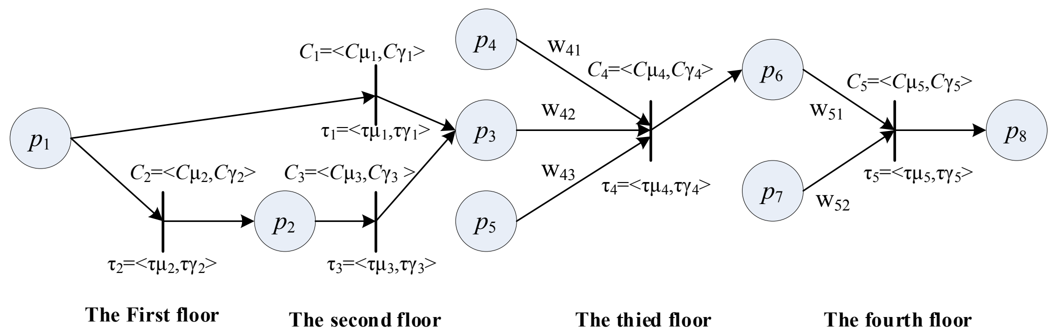


Figure 2: Layered model of intuitionistic fuzzy Petri net

4.1 Model hierarchy

First, we divide it by algorithm 1. Because τ_1 and τ_3 correspond to the same output library p_3, Step 4 removes t_1 from the first layer and $T_1 \neq \varnothing$, so there is no need to add virtual libraries and virtual transitions, and only divides t_1 and t_3 into the same layer, as shown in Fig. 2. After stratification, $T_1 = \{t_2\}$, $T_2 = \{t_1,\ t_3\}$, $T_3 = \{t_4\}$, $T_4 = \{t_5\}$, the continuous function expression of each library is as follows:

Simplicity makes $M_\pi = 0$

(1) Ignite the transition t_2 in $T_1, x_\mu = M_\mu(p_1), x_\gamma = M_\gamma(p_1)$, $\tau_\mu = \tau\mu_2, \tau_\gamma = \tau\gamma_2$, so:

$$\begin{cases} M_\mu(p_2) = M_\mu(p_1)C_2 / [(1+e^{-b(M_\mu(p_1)-\tau\mu_2)})(1+e^{b(M_\gamma(p_1)-\tau\gamma_2)})] \\ M_\gamma(p_2) = 1 - M_\mu(p_1)C_2 / [(1+e^{-b(M_\mu(p_1)-\tau\mu_2)})\cdot(1+e^{b(M_\gamma(p_1)-\tau\gamma_2)})] \end{cases}$$

(2) Ignite the transition t_1 and t_3 in T_2,

let: $M1_\mu(p_3) = M_\mu(p_1)C_1 / [(1+e^{-b(M_\mu(p_1)-\tau\mu_1)})(1+e^{b(M_\gamma(p_1)-\tau\gamma_1)})]$, $M1_\gamma(p_3) = 1 - M1_\mu(p_3)$,

$M2_{\mu}(p_3)=M_{\mu}(p_2)C_3/[(1+e^{-b(M_{\mu}(p_2)-\tau\mu_3)})(1+e^{b(M_{\gamma}(p_2)-\tau\gamma_3)})]$, $M2_{\gamma}(p_3)=1-M2_{\mu}(p_3)$,

so $M_{\mu}(p_3)=\vee(M1_{\mu}(p_3),M2_{\mu}(p_3)), M_{\gamma}(p_3)=\wedge(M1_{\gamma}(p_3),M2_{\gamma}(p_3))$, that is:

$$\begin{cases} M_{\mu}(p_3)=M1_{\mu}(p_3)/[1+e^{-b(M1_{\mu}(p_3)-M2_{\mu}(p_3))}]+M2_{\mu}(p_3)/[1+e^{-b(M2_{\mu}(p_3)-M1_{\mu}(p_3))}] \\ M_{\gamma}(p_3)=M1_{\gamma}(p_3)/[1+e^{b(M1_{\gamma}(p_3)-M2_{\gamma}(p_3))}]+M2_{\gamma}(p_3)/[1+e^{b(M2_{\gamma}(p_3)-M1_{\gamma}(p_3))}] \end{cases}$$

(3) Ignite the transition t_4 in T_3,

$x_{\mu}(t_4)=M_{\mu}(p_4)\times w_{41}+M_{\mu}(p_3)\times w_{42}+M_{\mu}(p_5)\times w_{43}$,

$x_{\gamma}(t_4)=M_{\gamma}(p_4)\times w_{41}+M_{\gamma}(p_3)\times w_{42}+M_{\gamma}(p_5)\times w_{43}$, $\tau_{\mu}=\tau\mu_4, \tau_{\gamma}=\tau\gamma_4$, so:

$$\begin{cases} M_{\mu}(p_6)=x_{\mu}(t_4)C_4/[(1+e^{-b(x_{\mu}(t_4)-\tau\mu_4)})(1+e^{b(x_{\mu}(t_4)-\tau\gamma_4)})] \\ M_{\gamma}(p_6)=1-x_{\mu}(t_4)C_4/[(1+e^{-b(x_{\mu}(t_4)-\tau\mu_4)})(1+e^{b(x_{\mu}(t_4)-\tau\gamma_4)})] \end{cases}$$

(4) Ignite the transition t_5 in T_4,

$x_{\mu}(t_5)=M_{\mu}(p_6)\times w_{51}+M_{\mu}(p_7)\times w_{52}$, $x_{\gamma}(t_5)=M_{\gamma}(p_6)\times w_{51}+M_{\gamma}(p_7)\times w_{52}$,

$\tau_{\mu}=\tau\mu_5, \tau_{\gamma}=\tau\gamma_5$, so:

$$\begin{cases} M_{\mu}(p_8)=x_{\mu}(t_5)C_5/[(1+e^{-b(x_{\mu}(t_5)-\tau\mu_5)})(1+e^{b(x_{\mu}(t_5)-\tau\gamma_5)})] \\ M_{\gamma}(p_8)=1-x_{\mu}(t_5)C_5/[(1+e^{-b(x_{\mu}(t_5)-\tau\mu_5)})(1+e^{b(x_{\mu}(t_5)-\tau\gamma_5)})] \end{cases}.$$

4.2 Training test

The ideal parameters of the hypothetical model are:

w_{41}=0.2, w_{42}=0.5, w_{43}=0.3, w_{51}=0.4, w_{52}=0.6, C_1=(0.7,0.25),
C_2=(0.9,0.08), C_3=(0.6,0.36), C_4=(0.8,0.15), C_5=(0.7,0.25),
τ_1=(0.3,0.65), τ_2=(0.4,0.56), τ_3=(0.2,0.78), τ_4=(0.5,0.5), τ_5=(0.4,0.56).

According to the IFPN reasoning algorithm, 200 training samples are generated to train the model. The constant of inference function is b=5000, and the initial learning rate is η=0.1. After that, the output error of the network is adjusted dynamically according to formula (10), ε=0.06. The original parameters of the model are initialized by random number before training. After 300 studies, the total average error function value is 0.058<0.06, which meets the requirements. The optimization results of each parameter are shown in Tab. 1.

Table 1: Parameter optimization results

Para-meter	Actual Output	Mean Square Error	Para-meter	Actual Output	Mean Square Error	Para-meter	Actual Output	Mean Square Error
w_{41}	0.1795	0.3291×10^{-3}	C_1	(0.6865, 0.2752)	2.6973×10^{-3}	τ_1	(0.3431, 0.6124)	2.6929×10^{-3}
w_{42}	0.4863		C_2	(0.7826, 0.1474)		τ_2	(0.4398, 0.4987)	
w_{43}	0.3335		C_3	(0.5191, 0.3713)		τ_3	(0.2529, 0.7297)	
w_{51}	0.3814		C_4	(0.8243, 0.1666)		τ_4	(0.5298, 0.4405)	
w_{52}	0.6201		C_5	(0.6519, 0.2881)		τ_5	(0.3531, 0.6128)	

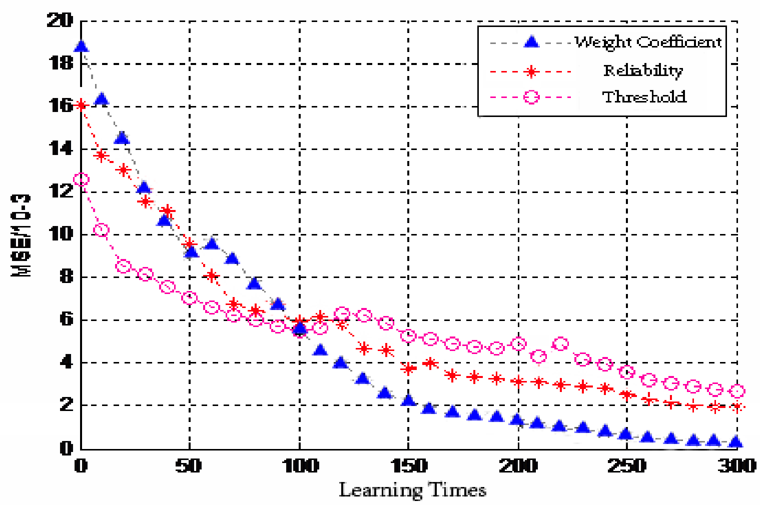


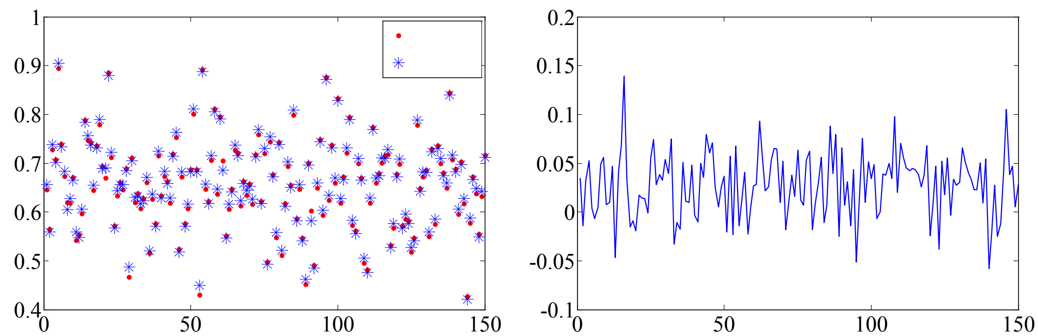
Figure 3: Parameter MSE change curve graph

Mean Square Error (MSE) is an index used to measure the accuracy of parameters. It refers to the expected value of the square of the difference between the estimated value of parameters and the true value of parameters. The smaller the value, the higher the accuracy. In this paper, the parameters of mean square error with the times of learning curve as shown in Fig. 3. Since the evaluation function of the training process is the sum of squares of the difference between the actual output and the expected output of the sample, it has no functional correlation with MSE, so it is normal for some oscillation phenomena to occur in the graph. It can be seen that the BP algorithm has a high accuracy in adjusting w, while it performs relatively general performance on the other two sets of parameters.

For the above optimization parameters, we must analyze the advantages and disadvantages of the optimization results from another angle, that is, generalization performance. The performance evaluation index APE of literature [Wu, Er and Gao (2001)] is used. It is defined as:

$$\mathrm{APE} = [\sum_{i=1}^{n}(|t_i - y_i| / |t_i|)] / n \tag{14}$$

Among them, n is the number of data samples; t_i and y_i are the i expected outputs and the actual outputs respectively. The smaller the APE value is, the stronger the generalization performance is. 150 test samples were taken and the IFPN formal reasoning method was applied to optimize the model parameters. The actual output of the termination library p_8 was obtained and compared with the expected output. As shown in Fig. 4, the performance of the APE index is compared with other similar methods, as shown in Tab. 2.



a) Comparison between expected and actual values b) Error between expected and actual values

Figure 4: Test results of IFPN model parameters

Table 2: Comparison between BP-IFPN method and other methods

Method	Weight MSE	Reliability MSE	Threshold MSE	Test Set APE
BP-FPN	0.6987×10^{-3}	5.5039×10^{-3}	2.7756×10^{-3}	0.0968
GA-FPN	0.41×10^{-3}	3.41×10^{-3}	4.75×10^{-3}	0.0718
ACA-FPN	0.3402×10^{-3}	2.3612×10^{-3}	2.6655×10^{-3}	0.0361
BP-IFPN	0.3291×10^{-3}	2.4623×10^{-3}	2.7034×10^{-3}	0.0349

Thus, the output of the optimization model presented in this paper coincides basically with the expected value, that is, the parameter optimization learning algorithm has good generalization performance. Compared with other similar methods, the generalization performance of this model is also optimal. The performance indexes of parameters are better than those of BP-FPN method and GA-FPN method, and slightly inferior to ACA-FPN method. But ACA-FPN method is often given a lot of restrictions in practical application, and there are also some questions about whether it can be reasonably converted into ACA application, which make it not always converge to the true value. Similarly, GA has similar problems. Therefore, the BP-IFPN method is superior, but the disadvantage of this method is slow convergence. This paper solves this problem to a certain extent by introducing momentum and variable step size.

Because there are many parameters to be learned, the adjusted parameters obtained by IFPN converge to the local extremum of the input parameters, which is not consistent with the ideal parameters. As shown in Tab. 1, the threshold parameters are quite different. However, the parameters in IFPN are of special significance, which is the biggest difference between IFPN training and neural network training. By analyzing the validity of the adjusted parameters obtained after learning, the input parameters are continuously adjusted so that the values obtained are close to the ideal parameters.

5 Conclusions

In this paper, an intuitionistic fuzzy petri nets model is proposed by extending the classical BP algorithm intuitively. The algorithm adds non-membership parameter, which not only makes the description of uncertain information more precise, but also makes the optimization result more ideal because of the inverse effect of non-membership parameter. At the same time, momentum and variable step size are introduced to reduce the oscillation trend and improve the convergence. Therefore, this optimization model not only has the ability of efficient parallel processing, but also has the self-learning ability like BP network. It is of great significance to the establishment and maintenance of intuitionistic fuzzy production rule base system, and provides theoretical support for solving the model construction and reasoning problems of some complex system structures in the field of information fusion and is useful for information services. How to combine several algorithms to optimize high-precision system parameters by giving full play to their respective advantages will be the focus of future research.

References

Cai, Z.; Wang, Z.; Zheng, K.; Cao, J. (2013): A distributed TCAM coprocessor architecture for integrated longest prefix matching, Policy Filtering, and Content Filtering. *IEEE Transactions on Computers*, vol. 62, no. 3, pp. 417-427.

Cui, J.; Zhang, Y.; Cai, Z.; Liu, A.; Li, Y. (2018): Securing display path for Security-Sensitive applications on mobile devices. *Computers, Materials & Continua*, vol. 55, no. 1, pp. 17-35.

Li, X.; Felipe, L. (2000): Adaptive fuzzy Petri Nets for dynamic knowledge representation and inference. *Expert Systems with Applications*, vol. 19, no. 3, pp. 235-241.

Li, Y.; Le, X. (2007): Application of ant colony algorithms in parameter optimization of fuzzy Petri nets. *Computer Application*, vol. 27, no. 3, pp. 638-641.

Li, X.; Yu, W.; Felipe, L. (2000): Dynamic knowledge inference and learning under adaptive fuzzy petri net framework. *IEEE Transactions on System, Man and Cybernetics, Part C: Applications and Reviews*, vol. 30, no. 4, pp. 442-450.

Liu, F.; Guo, Y.; Cai, Z.; Xiao, N.; Zhao, Z. (2019): Edge-enabled disaster rescue: a case study of searching for missing people. *ACM Transactions on Intelligent Systems and Technology*, vol. 10, no. 8, pp. 1-17.

Liu, F.; Tang, G.; Li, Y.; Cai, Z.; Zhang, X. et al. (2019): A Survey on edge computing systems and tools. *Proceedings of the IEEE*, vol. 107, no. 8, pp. 1-26.

Liu, Y.; Liu, A.; Liu, X.; Huang, X. (2019): A statistical approach to participant selection in location-based social networks for offline event marketing. *Information Sciences*, vol. 480, pp. 90-108.

Liu, H.; You, J.; Li, Z.; Tian, G. (2017): Fuzzy Petri Nets for knowledge representation and reasoning: a literature review. *Journal Engineering Applications of Artificial Intelligence*, vol. 60, pp. 45-56.

Shen, X.; Lei, Y.; Li, C. (2009): Intuitionistic fuzzy Petri nets model and reasoning algorithm. *Sixth International Conference on Fuzzy Systems and Knowledge Discovery*, vol. 6, no. 3, pp. 119-122.

Sun, X.; Wang, N. (2018): Gas turbine fault diagnosis using intuitionistic fuzzy fault Petri nets. *Journal of Intelligent and Fuzzy Systems*, vol. 34, no. 6, pp. 3919-3927.

Tan, J.; Liu, W.; Xie, M.; Song, H.; Liu, A. et al. (2019): A low redundancy data collection scheme to maximize lifetime using matrix completion technique. *EURASIP Journal on Wireless Communications and Networking*, vol. 5.

Teng, H.; Liu, Y.; Liu, A.; Xiong, N. N.; Cai, Z. et al. (2019): A novel code data dissemination scheme for internet of things through mobile vehicle of smart cities. *Future Generation Computer Systems*, vol. 94, pp. 351-367.

Tan, J.; Liu, W.; Wang, T.; Xiong, N.; Song, H. et al. (2019): An adaptive collection scheme-based matrix completion for data gathering in Energy-Harvesting wireless sensor network. *IEEE Access*.

Tsang, E.; Yeung, D.; Lee, J. (1999): Learning capability in fuzzy petri nets. *IEEE. Proceeding of International Conference on Systems, Man and Cybernetics*, pp. 355-360.

Wu, S.; Er, M.; Gao, Y. (2001): A fast approach for automatic generation of fuzzy rules by generalized dynamic fuzzy neural networks. *IEEE Transactions on Fuzzy Systems*, vol. 9, no. 4, pp. 578-594.