IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received November 25, 2020, accepted November 29, 2020, date of publication December 3, 2020,
date of current version December 18, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.3042439

A Fuzzy Ensemble Method With Deep Learning for Multi-Robot System

PENG LIAO[1], **MG XU**[2], **AND CONGYING YANG**[2]
[1] Aerodynamics Theory and Application Institute, China Academy of Aerospace Aerodynamics, Beijing 100074, China
[2] Software Research and Development Department, Xi'an Bazhentu Network Technical Cooperation, Xi'an 710003, China

Corresponding author: Peng Liao (pengliao6666@163.com)

ABSTRACT In a multi-robot system, situation assessment evaluates the current situation quantitatively to help decision-makers make the best decision. Conventional situation assessment methods ignore the initiative of each robot, so it often encounters bottlenecks. Collaborative intelligence shows better performance than a single global decision. To address this problem, this work introduces a deep learning-based fuzzy adaptive method (DLFA) to achieve the real-time situation assessment for a multi-robot system. The proposed method employs the shortest path faster algorithm to achieve information sharing between agents. The shortest path faster algorithm ensures that the agent distributes its state information to its teammates in the fastest way. Each agent gets the information from teammates and treats their state as the observation of the scene. Deep neural network maps current observations into a local situation assessment result by combining a large number of nonlinear processing layers. Finally, each local assessment result is regarded as a brick to construct the final situation assessment via a fuzzy ensemble method. Experimental results show that the proposed method outperforms competitors.

INDEX TERMS Multi-robot system, situation assessment, shortest path faster algorithm, deep learning, fuzzy ensemble method.

I. INTRODUCTION

Many heterogeneous or isomorphic agents constitute a multi-agent system [1]. In the multi-agent environment, each agent perceives different information and perform different actions. A multi-agent system is a popular platform for the study of artificial intelligence technology, including reinforcement learning, deep learning, and machine learning [2], [3].

The existence of a large number of independent agents makes the multi-agent system complex, dynamic, and uncertain [4]. The potential impact of one agent on other agents should be of concern. Therefore, the research of a multi-agent system is often more attractive than a single agent [5], [6]. In cooperative tasks, each agent must cooperate with his teammates when facing competition. A good collaborative process is a key component. A perfect confrontation needs to be completed under the guidance of an appropriate attack strategy and a good cooperative strategy [7]. The premise of coordination and confrontation requires our side to evaluate the situation reasonably [8].

Situation assessment evaluates some factors in the current scene to predict the changes in the next stage [9]. It provides a systematic way for us to understand the situation between our side and the enemy. Then, an appropriate decision is made. To evaluate the current situation of the multi-agent system more effectively, some factors should be considered, such as our score, enemy score, and the position of both sides. It is critical to construct an effective situation assessment framework to correctly evaluate the factors [10], [11].

Conventional global assessment methods do not consider the influence of individual agents on the overall situation assessment results. For example, in a robot soccer game, our side always controls the ball, but rarely achieve the shooting behavior. This case may be judged to be a good situation for our side by conventional methods. It is likely to be an unwise assessment. So, this paper considers the collaborative evaluation and uses a fuzzy method to integrate the results of each agent to achieve a better assessment. The ensemble method is a good solution. Meanwhile, different agents give different influences on the final assessment results. This paper considers that different agents affect the results in different degrees.

Recently, some artificial intelligence methods have been leveraged to achieve situation assessments, such as the

The associate editor coordinating the review of this manuscript and approving it for publication was Shadi Alawneh.

Bayesian method [12], evidence theory [13], multiple linear regression [14], BP neural network [15]. The multiple linear regression method maps scene factors into a result of the situation assessment in a linear way. It is an effective method for the confrontation decision problem in simple scenarios. Many factors will bring more difficulties to situation assessment. The multiple linear regression method needs to consider considerable factors to produce a suitable rule, which is suffocating in practical application. The prior conditions for conducting the evidence theory are very strict, and considerable prior data are necessary. The conflict resolution dilemma between evidence has not been relaxed, which leads to an imprecise assessment of the current situation. The naive Bayes method requires a set of sample data to satisfy independent distribution. In [16], an auxiliary system has been developed to assist pilots to achieve reasonable air combat situation assessment using the Bayesian method and fuzzy logic [17]. In [18], a fuzzy clustering method is proposed to cluster the scene sample data, and then a semi-supervised Bayesian method is employed to partition the real-time situation scene. The Bayesian method requires a large number of independently distributed sample data for training. In the actual scenario, it is hard to get enough independently distributed samples. Therefore, this method is not practical for some complex scenes.

Neural networks provide a solution for generating a nonlinear mapping from scene factors to assessment results. The BP neural network extracts features of sample data via a cascade between layers to learn a mapping from scene factors to situation assessment results [19]. This method is effective to achieve feature representation and mapping learning, simultaneously. Limited state representation and feature extraction become a bottleneck, which restricts the performance of BP methods. Deep learning allows the computational model to be composed of considerable cascading processing units to learn the deep representation of data. This approach has gone beyond many start of art approaches in robotics and non-robotics. The convolutional neural network (CNN) recently has become a popular technology and shows excellent performance for many real applications [20], [21]. The parameter sharing of the convolution kernel and sparse connection between layers allows the CNN network to achieve effective feature extraction without high computational complexity.

Previous work has proposed a lot of structures of CNN for different tasks [22], [23]. The convolution layer is a key component of CNN. A single partition in the input space can trigger a stimulus for each neuron in the convolution layer. CNN can be applied to many supervised learning tasks with labels, such as classification tasks and prediction tasks. A major challenge is to train deep learning models using small-scale samples because it is expensive to get enough training data. Limited labeled samples bring great difficulties for the deep learning model. One feasible method is to construct a simple model to achieve classification at the expense of execution efficiency. Inspired by ensemble learning, several simple CNN computing models are aggregated to achieve a more advanced execution effect in some way. Each CNN model is trained in its way, and each one is treated as a brick to build the final building of situation assessment. This divides and conquers approach expands the source of the learning experience to slightly alleviate the requirement with low model complexity.

It is worth mentioning that previous work focused on deep learning in the development of making strategy for a multi-agent system. However, few works have used the deep learning model to achieve situation analysis and assessment. So, this paper focuses on the research of deep learning for situation assessment because it is of great significance as a prerequisite for effective decision-making. A good fusion method can not only aggregate the evaluation results but also does not require high computation or learning costs. The fuzzy method may be a good way for fusion. A standardized method is needed to deal with the scene data from different sources without a uniform dimension.

Each agent transmits its state information to other agents in the task scene, and each agent makes decisions after receiving the information from other agents. Different data transmission strategies bring different efficiency of information sharing for a multi-agent system. A network communication model allows this information sharing between different agents. In [24], the authors proposed a distribution-based method of information exchange for rotorcraft groups. Multiple UAVs share each other's information to achieve formation flying. In [25], multi UAVs optimize information sharing efficiency via a network optimization algorithm is investigated. Each agent calculates the velocity of other agents according to the received information. Relative velocities are used to evaluate the transmission quality of the network. The clustering algorithm is an alternative way to achieve the equivalent network communication for a group agent. Multiple agents with similar positions are divided into a cluster. The agents are divided into a cluster to achieve local communication with each other, and the central nodes of each cluster realize global communication. This method reduces the loss of the original direct point-to-point communication of all agents. However, new clustering is needed once the position of the agent changes, which shows that the effect of this method is short-lived.

The available communication strategies are developed for different communication tasks, and the cost is that the multi-agent system must bear a considerable amount of computation. Link congestion is fatal for real-time communication [26]. More computing sources being wasted in the communication layer means fewer resources are allocated to the evaluation and decision layers. For general multi-agent systems, such as robot soccer games, the minimalist principle gives the inspiration to focus on more critical components and minimize the loss of other components [27]. The shortest path faster algorithm is employed to achieve point-to-point communication for situation assessment. The awareness of information sharing is achieved among participants via the shortest path faster algorithm to facilitate the situation assessment process.

In this study, a deep learning-based adaptive situation assessment method is proposed for a multi-agent system. In the communication layer, the shortest path faster algorithm is used to develop the communication network model between agents. Each agent can share its status information with other agents through the communication network. This method reduces network congestion compared with the original multi-source transmission. A standardization method eliminates the dimensional differences between different scene data. Then, a CNN based deep learning model allows different agents to predict the current results for situation assessment according to their information. Different prediction results are fused by a fuzzy method. This fuzzy fusion method produces the final result. The result of the situation assessment guides us to produce the best action and get positive feedback.

The main contributions are as follows.

1) A shortest path faster algorithm is used to achieve information sharing among agents. Each agent can provide its observation to teammates in the fastest way.
2) For the situation assessment, a deep learning model is employed to give a local evaluation result for each agent. Each agent inputs its observation into the deep learning model to get a local situation assessment result.
3) Then, a fuzzy fusion method is proposed to aggregate the local evaluation results of each agent. This approach uses collective wisdom, which is more effective than a single global decision.

The organization of this work is as follows. Section II describes the baseline approaches. Section III presents the proposed method including the communication strategy, deep learning-based situation assessment method, and the fuzzy fusion method. Experiments are conducted to demonstrate the effectiveness of the proposed methods in Section IV. The experiment includes two different simulations of the robot soccer game. Conclusions are drawn in the last section.

II. BACKGROUND

A. SITUATION EVALUATION METHOD BASED ON MULTIPLE LINEAR REGRESSION (MLR)

The linear regression method is introduced into the situation assessment model. The linear regression method combines the scene factor and the situation evaluation results via expert experience. Then, fitting training is conducted. Multiple linear regression (MLR) has been developed for the task with multiple evaluation factors. The procedure for the MLR is as follows:

Step1: The labels for the situation assessment are given using the expert experience and the set for labels is $\{L_1, L_2, L_3, \ldots, L_m\}$. Each label for the situation assessment is given by,

$$L_i = i, \quad i = 1, 2, \ldots, m \tag{1}$$

Step2: The scene data within a certain period is got from the actual scene. There are K evaluation factors, so the MLR model has K independent variables. K evaluation factors have different dimensions, and we need to unify the dimensions of these factors. We can set the range of the value for each evaluation factor according to the collected data.

Step3: MLR model for situation assessment was trained using sample data. Then, the model outputs the final results for the situation assessment.

A multi-agent system often faces a dynamic and complex decision-making environment. A feasible solution is to simplify the nonlinear mapping relationship into a linear one with some constraints, which reduces the calculation of the model. But, a naive linear relation cannot describe the mapping from observation space to decision space. The bad mapping relationship gives a bad decision, which often leads to our loss in the game. There is a nonlinear relationship between the situation results and the factors, and the MLR method is not necessarily able to satisfy the real-time situation assessment. The neural network provides a nonlinear fitting function to address this problem.

B. SITUATION EVALUATION METHOD BASED ON BP NEURAL NETWORK (BP-NN)

Another conventional situation assessment method is the BP neural network. The topological structure of the BP neural network model is a three-level network, including a single input layer, multiple hidden layers, and a single output layer. In the output layer, the sigmoid function ensures that the output is kept in a small range. The basic process of situation assessment method using BP neural network is as follows:

Step 1: A BP neural network model is defined. After the reasonable activation function is selected, the neural network model from the input layer to the output layer is defined. $A_i = \left\{a_1^i, a_2^i, a_3^i, \ldots, a_M^i\right\}, i = 1, 2, 3, .., N$ is input vector and $N$is the number of training samples. $B_i = \{f_1(n), f_2(n), f_3(n), \ldots, f_N(n)\}$ is the output for the $n-th$ iteration.

Step 2: The label set for the situation assessment is defined. To describe the scene reasonably, the situation data of the real-time scene is given. The labels for the situation are $F = \left\{\hat{f}_i | i = 1, 2, 3, \ldots, N\right\}$.

Step 3: The training samples are used to train the BP model. The training sample is $A = \{A_1, A_2, \ldots, A_N\}$. Each data for the factor of the situation assessment is normalized. The normalized data is the label for the situation, which is given by,

$$\begin{aligned}\hat{A} = \Big\{&\left\{a_1^1, a_2^1, \ldots, a_m^1, f_1\right\}, \left\{a_1^2, a_2^2, \ldots, a_m^2, f_2\right\}, \ldots, \\ &\left\{a_1^N, a_2^N, \ldots, a_m^N, f_N\right\}\Big\}\end{aligned} \tag{2}$$

Step 4: Forward Computing. The output of the BP network is got, and the error of the sample is calculated.

Step 5: Back-propagation. The weights are updated using the error. If the termination condition is met, the learning process ends. Otherwise, the learning process will continue.

Step 6: Achieve real-time situation assessment. When the new data enters the BP-based situation assessment model, the neural network calculates the result for each label of the situation.

The BP-based situation assessment model has a limited capacity for feature representation and cannot extract deep-seated features. Deep learning can provide a more advanced computing model.

III. THE DEEP LEARNING BASED ADAPTIVE SITUATION EVALUATION METHOD

This work uses several local situation assessment results. This method allows each agent to get situation assessment results according to their observation, instead of a simple global situation assessment. An agent receives the state information from other agents as observation data for situation assessment. Each agent will get a local result for situation assessment using the deep learning computing model. A fuzzy method is developed to fuse the local situation assessment results of each agent to get the final result. In the situation assessment model, the shortest path faster algorithm ensures that agents can transmit their state information to each agent in a fast way.

A. MULTI-AGENT INFORMATION SHARING METHOD WITH A SHORTEST PATH FASTER ALGORITHM

For the situation assessment model, each agent transmits information to each other. Each node can transfer its state information, such as position, velocities, and so on, to achieve information sharing between agents. The obtained state data is used as the input of the situation assessment model. The information-sharing structure between agents is shown in Fig.1. In the end-to-end communication between agents, how to choose the optimal transmission path of the multi-agent system requires an efficient data path selection algorithm. The shortest path faster algorithm (SPA) is a single source shortest communication optimization algorithm, which can still work effectively when the graph structure contains negative weight edges. This algorithm is used to achieve information sharing among agents.

The procedure of the information sharing based on the shortest path faster algorithm for the multi-agent is as follows.

Step 1: Initialization. The queue for the central agent is initialized as *Quene*, and the set for the other agents is T. For the other agents, if they are connected with the central agent, the distance *Dis* is 1, otherwise, it is ∞.

Step 2: Update distance. The agent in *Quene* is used as the relay agent to update the shortest distance *Dis* from each agent in the set T to the central agent. Algorithm 1 gives the shortest distance update process for information sharing.

Step3: Update the queue. The agents with the shortest distance in the set T join the *Quene*, and the corresponding relay agents are out of the *Quene*.

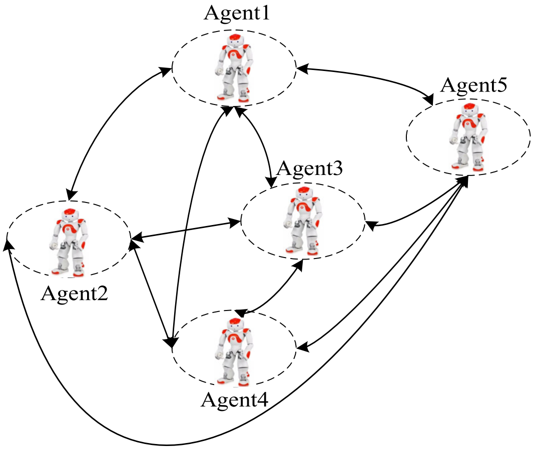

FIGURE 1. **The structure for information sharing between agents.**

Algorithm 1 Distance Update Process for Information Sharing

```
Definition
S =: The source node, that is the central agent node
T =: other agent nodes
Q =: Queue for storaging relay nodes
Distance =: The shortest distance from the source node
    to the remaining nodes
JoinQuene() =: Join queue
LeaveQuene() =: Leave the queue
Initialization
Repeat (for each step)
If $Dis\tan ce\,(i, m)+Dis\tan ce\,(m, j) < Dis\tan ce\,(i, j)$,
    $i \in S, m \in Q, j \in T$
    Then $Dis\tan ce\,(i, j) = Dis\tan ce\,(i, m) + Dis\tan ce\,(m, j)$ //Update the shortest distance
    $JoinQuene\,(Q, j)$
    $LeaveQuene\,(Q, k)$
    $t++$
    Until $Q$ is empty
```

Step4: Repeat the above iterations. Repeat steps 2 and 3 above until the *Quene* is empty.

B. STANDARDIZATION FOR THE SCENE DATA

Fig.2 shows the structure for the proposed method. A global neural network is developed due to each agent has similar input. Different agents input their scene factors into the deep learning-based computing model to get the local situation evaluation results. In this way, experience sharing reduces storage loss. When the local situation assessments are achieved, a fuzzy method fuses the local situation assessment results of each agent to get the global result. In practice, the global situation assessment result is needed for real-time task scenarios, while the local situation assessment is a replaceable way. We need to use the overall situation assessment results to make decisions, and the local situation assessment results can not describe the complete current situation.

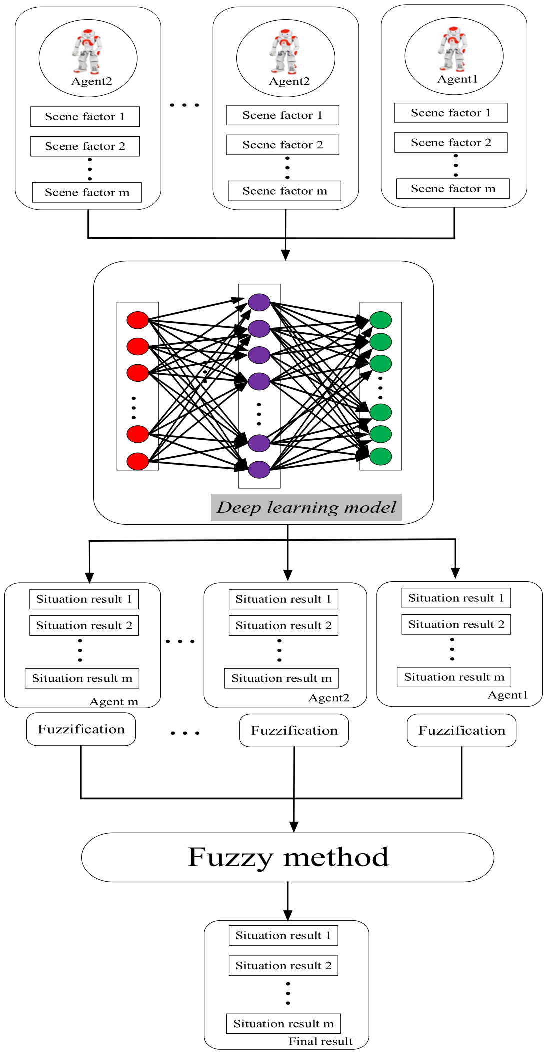

FIGURE 2. The structure for the proposed method.

So, the global situation assessment information is preferred for the multi-agent system. The proposed method is robust to this case when the agents can not continue to communicate. That is, the global situation assessment results can not be obtained, the local situation assessment is an alternative.

The collected scene data has different dimensions, so it needs to be standardized. The method for standardization is shown in Eq. (3) and Eq. (4). The scene data is recorded and the range of the scene data is obtained. In Eq. (3), for evaluation factor i, the maximum and minimum values $p_i^{\min}, p_i^{\max}$. of each scene data are recorded and calculated to normalize the range of scene data $\left[p_i^{down}, p_i^{up}\right]$, which is shown in Eq. (4).

$$p_i^{down} = \begin{cases} \alpha p_i^{\min}, p_i^{\min} \geq 0 \\ \beta p_i^{\min}, p_i^{\min} < 0 \end{cases} \tag{3}$$

$$p_i^{up} = \begin{cases} \alpha p_i^{\max}, p_i^{\max} < 0 \\ \beta p_i^{\max}, p_i^{\max} \geq 0 \end{cases} \tag{4}$$

where α, β are constants respectively, which are set according to the specific situation. For each value of the evaluation factor i, the normalized value p_i' is as follows:

$$p_i' = \begin{cases} \dfrac{p_i' - p_i^{\min}}{p_i^{\max} - p_i^{\min}}, & p_i^{\min} \leq p_i' < p_i^{\max} \\ 1, & p_i' \geq p_i^{\max} \\ 0, & p_i' < p_i^{\min} \end{cases} \tag{5}$$

C. THE DEEP LEARNING-BASED SITUATION ASSESSMENT

Each agent collects scene data, and then uses deep learning to achieve the local situation assessment. Momentum and Adam are used to improve the Back Propagation of neural networks [28]. The structure of the neural network for situation assessment based on deep learning is shown in Fig.3.

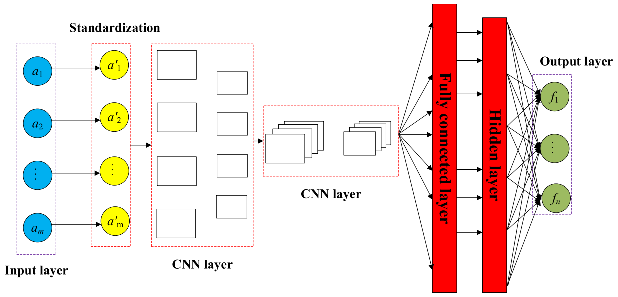

FIGURE 3. The structure of a neural network for deep learning.

The input for the neural network is $A = \{a_i | i = 1, 2, \ldots, M\}$ and label for the output is $\hat{F} = \left\{\hat{f}_i | i = 1, 2, \ldots, N\right\}$. The error between the actual output $F = \{f_i | i = 1, 2, \ldots, N\}$ and the label output is used to update these weights so that the current output is close to the expected value. The weights and biases are determined to minimize the error function. The adaptive momentum accelerates the convergence rate for the neural network, and restrain the possible oscillation of weight update. Momentum and Adam make the learning algorithm escape from the local optimal solution in the back-propagation process with a certain probability.

The process for the neural network model of the DLFA method is defined. The output of the hidden layer is given by,

$$h_1^i = \varphi\left(\sum\nolimits_{j=1}^{m} a_j \omega_1^{ij} + \delta_1^i\right), \quad i = 1, 2, 3, \ldots, M \tag{6}$$

The input signal continues to forward until the output layer.

The output for the neural network is given by,

$$f_k = \varphi\left(\sum\nolimits_{j=1}^{N} h_1^i \omega_2^{kj} + \delta_1^k\right), \quad k = 1, 2, 3, \ldots, N; \quad i = 1, 2, 3, \ldots, M \tag{7}$$

where φ is the sigmoid function. The error function is given by,

$$E(\boldsymbol{\omega}, \varphi) = \sum\nolimits_{k=1}^{N} \left(\hat{f}_k - f_k\right)^2 \Big/ 2 \tag{8}$$

After forward computing is completed, the back-propagation process is performed. The gradient descent method is used to

Algorithm 2 Procedure for the Improvement of the Back-Propagation Algorithm

```
Definition
    Learning rate and parameters: $\varepsilon$, $\eta$;$\gamma_1$, $\gamma_2$.
    Network weights $\omega$.
Initialization
  Repeat while is not terminal
    Sampling $m$ samples $\{a_1^i, a_2^i, \ldots, a_m^i, f^i\}$ from data set;
    Compute the output using forward computing;
    Get the error function $\nabla_\omega E(\omega_{t-1})$;
    Update the weights $\omega$ with the following process.
      $g_t = \gamma_2 g_{t-1} + (1-\gamma_2)\nabla_\omega E(\omega_{t-1})^2$;
      $m_t = \gamma_1 m_{t-1} + (1-\gamma_1)\nabla_\omega E(\omega_{t-1})$;
      $\hat{m}_t = \frac{\hat{m}_t}{1-\gamma_1^t}$;
      $\hat{g}_t = \frac{\hat{g}_t}{1-\gamma_2^t}$;
      $\omega_t = \omega_{t-1} - \frac{\eta\hat{m}_t}{\sqrt{\hat{g}_t+\varepsilon}}$;
Until neural network convergences.
```

update weights. In practical application, the weight update is related to the update episodes and is determined by its gradient change trend. The slow convergence rate of the neural network can be avoided by Momentum and Adam. Therefore, Momentum and Adam are introduced into the back-propagation process of the neural network. The improved back-propagation process is shown in algorithm 2. Algorithm 2 shows a new way to update weights using $\nabla_\omega E(\omega_{t-1})$. In this way, the convergence rate of the network can be accelerated, and the possible oscillation of weight updating can be restrained. This back-propagation process can also have a certain probability to get rid of the local optimal solution.

We use the training set to train the deep learning computing model, and then use the test set to test the output of the deep learning model. Training set and test set samples are collected from real-time confrontation scenarios.

D. THE FUZZY FUSION METHOD

In the actual confrontation task for multi-agent systems, the simple situation assessment results can not adapt to the complex situation. The fuzzy method shows excellent performance in handling nonlinear systems. A fuzzy method fuses these results of the local situation assessment using the deep learning model.

The process for the global situation assessment using a fuzzy method is as follows:

Step 1: Set the labels for the global situation assessment. The situation label of each agent is the same. The set for the labels of a situation assessment is F, which is given by,

$$F = \{f_i | i = 1, 2, 3, \ldots, N\} \tag{9}$$

Step 2: Get the results for the local situation assessment. The result for the situation assessment of the agent i is $F^i = \{f_{ij} | j = 1, 2, 3, \ldots, N\}$, N represents the number of labels for the situation.

Step 3: Define the fuzzification for the local situation assessment. The real-time situation assessment is fuzzed. The set of the local results using a fuzzification is given by,

$$\hat{F}^i = \left\{\hat{f}_{ik} | \hat{f}_{ik} = \frac{f_{ik}}{\sum_{k=1}^{n} f_{ik}}, k = 1, 2, 3, \ldots, N\right\} \tag{10}$$

For the agent i, the set for the fuzzy local results is $\hat{F}^i$.

Step 4: Define the fuzzy relation matrix. The fuzzy local results of each agent are got. The membership degree of the local result on the label is got, and the evaluation result is $f_i = \left(\hat{f}_{i1}, \hat{f}_{i2}, \hat{f}_{i3}, \ldots, \hat{f}_{iN}\right)$. The fuzzy relation matrix between each agent and the situation label is given by,

$$F_f = \begin{bmatrix} \hat{f}_{11} & \hat{f}_{12} & \hat{f}_{13} & \cdots & \hat{f}_{1N} \\ \hat{f}_{21} & \hat{f}_{22} & \hat{f}_{23} & \cdots & \hat{f}_{2N} \\ \hat{f}_{31} & \hat{f}_{32} & \hat{f}_{33} & \cdots & \hat{f}_{3N} \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ \hat{f}_{i1} & \hat{f}_{i2} & \hat{f}_{i3} & \cdots & \hat{f}_{iN} \end{bmatrix} \tag{11}$$

Step 5: Define the relative importance matrix for agents. It is necessary to determine the relative importance of each local result to the final situation assessment result. The relative importance of local situation assessment results reflects the relative importance of agents. The relation between the value for the important scales and the description is given in Table 1. P1, P2, P3, and P4 are empirical parameters. The partition for the important scales is shown in Fig.4.

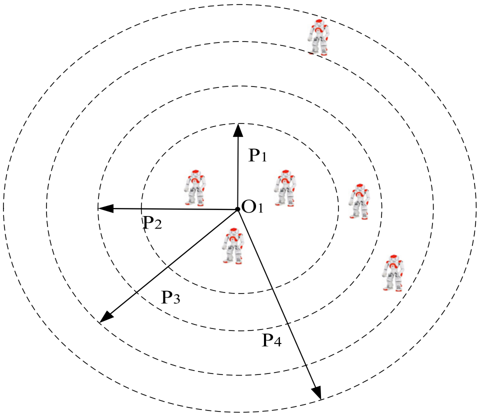

FIGURE 4. The division of importance scales for the agent.

TABLE 1. The important scales for the agent.

Description	Very important	Important	General important	Equal
Important scales	P1	P2	P3	P4

In Fig.3, we divide the scene into different regions P1, P2, P3, and P4, with the task target as the center and circles with different radii as the boundary. For example, in a soccer robot system, the center is the position of the enemy goal, and the scene area is divided by the distance between the robot and the enemy goal. Shooting the ball into the opponent's goal will give us points so the assessment result of the agent close to the goal is very important.

After the situation area is divided, the relative importance of agents is defined. The relative importance between the two agents is "equal" if these agents are in the same area. The agent near the center is "general important" to the agent outside if the two agents are located in the adjacent area. The agent near the center is "important" to the outside agent if two agents are located in the interphase region. The agent near the center is "very important" to the remote agent if the two agents are located in the far area. Then, the relative importance of agent i to the agent j is d_{ij}. A relative relation matrix **D** is defined to represent the relative importance between agents, which is given by,

$$\mathbf{D}=\begin{bmatrix} 1 & d_{1,2} & \cdots & d_{1,M-1} & d_{1,M} \\ \frac{1}{d_{1,2}} & 1 & \cdots & d_{2,M-1} & d_{2,M} \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ \frac{1}{d_{1,M-1}} & \frac{1}{d_{2,M-1}} & \cdots & 1 & d_{M-1,M} \\ \frac{1}{d_{1,M}} & \frac{1}{d_{2,M}} & \cdots & \frac{1}{d_{M-1,M}} & 1 \end{bmatrix} \tag{12}$$

Each data in Eq. (12) is processed using a fuzzy analytic hierarchy process and the processed data is given by,

$$\hat{d}_{ij}=\frac{\sum_{l=1}^{M} d_{il}}{2N}-\frac{\sum_{l=1}^{M} d_{jl}-N}{2N}i, \quad j=1,2,3,\ldots,M \tag{13}$$

Then, each row of the new matrix $\mathbf{D}=\left[d_{ij}\right]_{M\times M}$ is processed with Eq.(20) to get the fuzzy weight vector $(\omega_1,\omega_2,\ldots,\omega_M)$.

$$\omega_i=\sum_{j=1}^{M}\frac{1}{2N^2}\left(\sum_{l=1}^{M} d_{il}-\sum_{l=1}^{M} d_{jl}+N\right)+\frac{1}{N} \tag{14}$$

Step 6: Obtain a global situation assessment result. The global results are got using the maximum and minimum method, which is given by,

$$f_i=\vee_{l=1}^{N}\left(\omega_i \wedge_{i=1}^{M} f_{il}\right), \quad i=1,2,3,4,\ldots,M \tag{15}$$

The final result is $\hat{f}=Max\,(f_1,f_2,\ldots,f_M)$.

E. THE WHOLE ALGORITHM

The procedure for the proposed method is shown below. Firstly, a set for the scene factor is defined. The scene factors contain the elements affecting the robot situation assessment. The set for the scene factor is $A=\{a_i|i=1,2,\ldots,M\}$. The scene data corresponding to the scene factors is obtained in the multi-agent game. Then, the evaluation set is defined. The situation evaluation set is $F=\{f_i|i=1,2,3,\ldots,N\}$. Five situation labels are defined, which are "favorable, relatively favorable, general, relatively unfavorable, and unfavorable". After that, a neural network model for deep learning is developed. Momentum and Adam are introduced into the back-propagation process to satisfy the real-time situation assessment. The initial weights for deep learning are randomly assigned. A shortest path faster algorithm is employed to achieve information sharing between agents. The agent transmits its state information to other agents.

The standardized data is inputted into the neural network model and the local situation assessment is got. The computing process uses the state information of itself and other agents. A fuzzy method fuses the local situation assessment results. The results of local situation assessment are fuzzified, and the fuzzy relationship from local situation assessment to global situation assessment is designed. Finally, the global situation assessment result for a multi-agent system is obtained. Algorithm 3 presents the whole algorithm for the deep learning-based fuzzy situation evaluation for multi-agent systems.

Algorithm 3 The Whole Algorithm for the Proposed Method

```
Definition
    x = Number of agent
    $a_i$ := The i-th input scene data
    $a_i'$ := Standardized input data
    Samples: = A training sample with situation labels and scene data.
    Num_Samples: = The number of training data
    Deep_Net: = Neural network mode for Deep learning
    Normalization( ): = Standardization for input data
    Train_Inlayer( ): = Training of hidden layers of neural network
    Train ( ): = Training for deep learning
    Out ( ): = Computing the results
    Fuzzy(): = Fuzzification of output results.
    Training for the deep learning model:
    $i \leftarrow 0$;
Repeat $i++$
    $\hat{a}_i$ ←Normalization( Samples );
    Train_Inlayer(Net_Fc, $a_i'$ );
Until $i$ >Num_Samples;
Train (Deep_Net, Samples);
Computing of deep learning:
    $t \leftarrow 0$;
Repeat $t++$
    $a_i'$ ←Normalization($I_i$);
    $f_i$ ←Out ($a_i'$, Deep_Net);
    $\hat{f}_i$ ←Fuzzy($f_i$);
Until $t < T\_\max$
Until Done with all the agents
```

IV. EXPERIMENTS AND ANALYSIS

A. EXPERIMENT SETUP

Experiments are performed on the robot soccer simulation platform. The FIRA (Federation of International

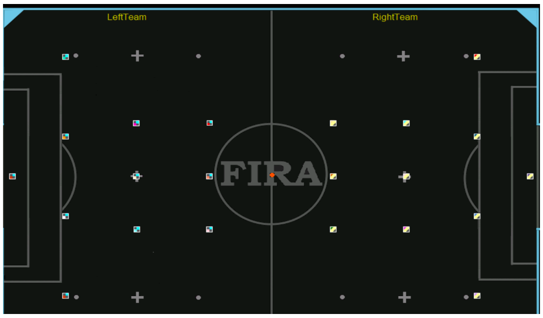

FIGURE 5. The simulation platform.

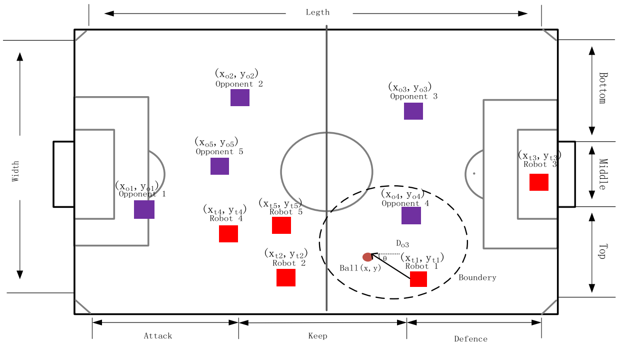
Legth
Width
Bottom
Middle
Top
(x_{o2}, y_{o2}) Opponent 2
(x_{o3}, y_{o3}) Opponent 3
(x_{o5}, y_{o5}) Opponent 5
(x_{o1}, y_{o1}) Opponent 1
(x_{t4}, y_{t4}) Robot 4
(x_{t5}, y_{t5}) Robot 5
(x_{o4}, y_{o4}) Opponent 4
(x_{t3}, y_{t3}) Robot 3
D_{o3}
θ
Ball (x, y)
(x_{t1}, y_{t1}) Robot 1
(x_{t2}, y_{t2}) Robot 2
Boundery
Attack
Keep
Defence

FIGURE 6. The structure of the platform.

Robot-soccer Association) [29] 11-vs-11 simulator is used for this experiment as a multi-agent system, as shown in Fig.5-6. The competitors include Multiple linear regression (MLR) [12], BP neural network method [15], and a Bayesian method [14]. Scene factors are the premise for the situation assessment. The score difference between the two sides, the Handing rate of our side, the Handing rate of a single agent, and the Shooting rate are selected as the scene factors.

The Handing rate $C$for N time steps is given by,

$$C = \sum_{j=1}^{N} \sum_{i=1}^{11} f(x_i, y_i) \Big/ \Big(C_o + \sum_{j=1}^{N} \sum_{i=1}^{11} f(x_i, y_i) \Big) \tag{16}$$

where C_o is the handing times of opponent. $f(x_i, y_i) = 1$ means our player i is handing ball, which is given by,

$$f(x_i, y_i) = \begin{cases} 1, \sqrt{(x_i - x_b)^2 + (y_i - y_b)^2} \leq R \\ 0, \sqrt{(x_i - x_b)^2 + (y_i - y_b)^2} > R \end{cases}; i = 1, 2, .., 11 \tag{17}$$

where (x_b, y_b) is the position of the ball. If the distance between the robot and the ball is less than a threshold R, it means that the robot is handing the ball. Eq. (17) indicates whether our robot hands the ball or not. The Handing rate is divided into the Handing rate of our side and the Handing rate of a single agent.

Shooting rate is given by,

$$\xi = {}^{U_t}\!/_{(U_t + U_o)} \tag{18}$$

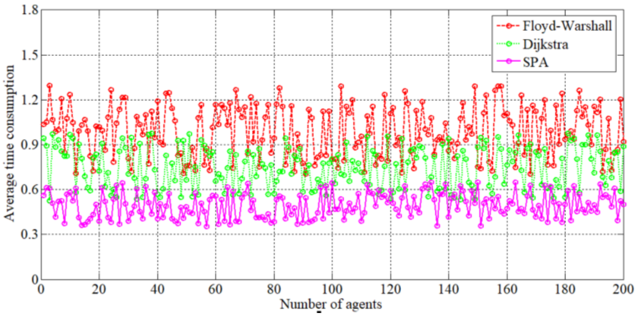

FIGURE 7. The simulation result with 200 robots.

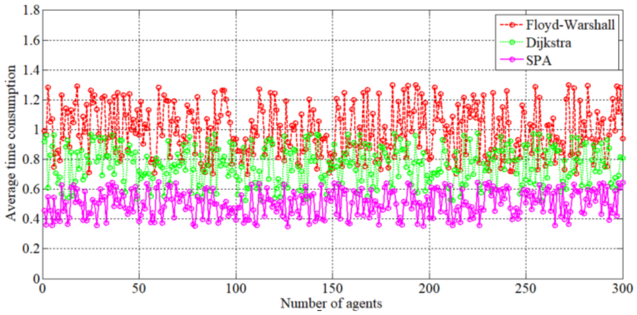

FIGURE 8. The simulation result with 300 robots.

where U_t is shooting times for our players and U_o is the opponent.

Some scene information needs to be achieved via the information sharing between agents, such as the handling rate of our side. Agents need to transmit their location information to other agents to know whether our agents are controlling the ball.

B. EXPERIMENT ON THE KNOWLEDGE SHARING

Real-time is an important metric for a multi-agent system. Some renowned multi-agent information-sharing methods are compared, including the Floyd-Warshall method [30] and the Dijkstra method [31]. The performance and effectiveness of the SPA are verified by this experiment.

Firstly, 100 large-scale weakly connected maps are generated randomly, each of which has a certain number of robots. Whether robots can communicate with each other and the cost is random. Then, the Floyd-Warshall method, SPA method, and the Dijkstra method are used to simulate the information sharing process. One robot will share its state information with all other robots via path selection. We randomly generated 200, 300, and 400 robots for each map for testing, and the average time consumption on sharing information was recorded. Finally, the average time consumption for the information sharing of the three methods was compared. The experimental results are shown in Fig. 7 to Fig. 9.

In Fig.7, when the Floyd Warshall method achieves information sharing in 100 randomly generated maps, the range for average time is 0.9s to 1.5s. When the Dijkstra method achieves information sharing in these 100 generated maps, the

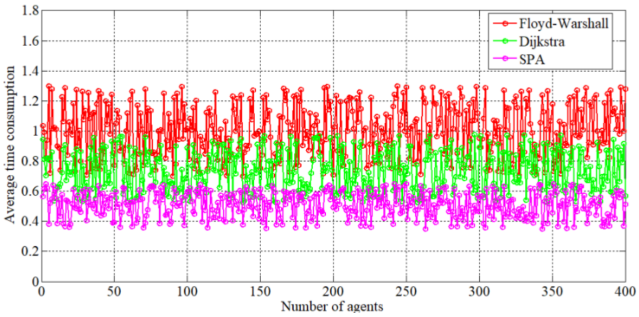

FIGURE 9. The simulation result with 400 robots.

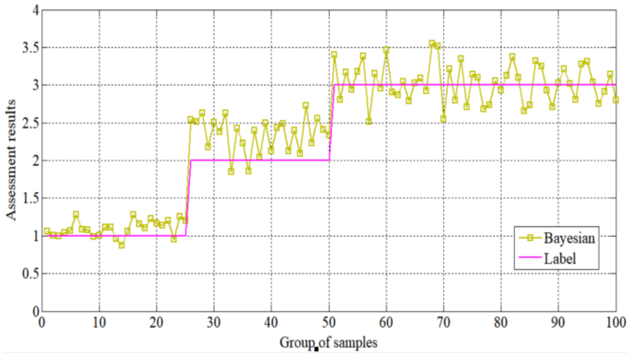

FIGURE 10. The training result using the Bayesian training method.

range for average time is 0.6s to 1.0s. When the SPA method performs information sharing in these 100 generated maps, the range for average time is 0.3s to 0.6s. SPA method has the shortest average running time. The average running time of the SPA method is about 42% shorter than the Floyd Warshall method and about 14% shorter than the Dijkstra method.

Similarly, Fig.8 and Fig.9 show similar experimental results. As the number of robots increases, the average running time of the SPA method is still the shortest. Floyd-Warshall method can search the shortest path between each robot and other robots in the sharing process. It is a multi-source shortest path search method, so it runs the longest. SPA method is a single-source shortest path search method, so it runs faster and takes a shorter time than the Floyd-Warshall method. The Dijkstra method will traverse all nodes gradually, so it is inefficient, although it is a common multi-agent information sharing method. SPA method has a shorter running time than Floyd-Warshall and Dijkstra methods in a weak connectivity environment. In conclusion, the SPA method has better efficiency and a shorter running time.

C. EXPERIMENT ON SITUATION ASSESSMENT

In the experiment, 200 groups of samples with the situation label are collected from the simulation platform. For the collected samples, 180 groups are taken as the training set. 20 groups are selected as the test set. In the training period, we randomly selected 100 groups of the training samples to compare the training effects of the proposed method and the renowned Bayesian training method [14]. After the training, in the test period, we test three different methods, including the proposed method, BP-NN method, and the MLR method via the test set. The efficiency of different situation assessment methods is evaluated. The experimental results for the training set are shown in Fig.10 to Fig.11. In the training phase, the situation assessment labels are sorted from large to small using their actual value to clearly show the experimental results.

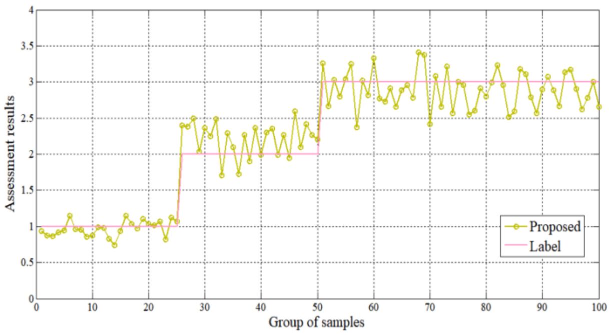

FIGURE 11. The training result using the proposed method.

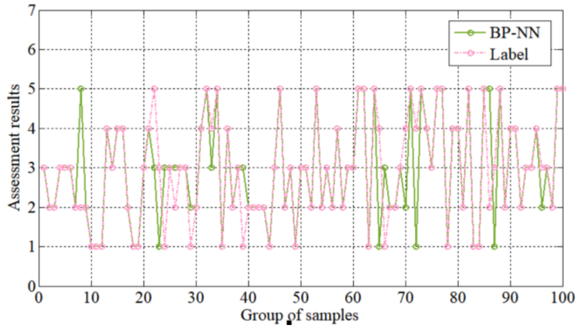

FIGURE 12. The assessment result with the BP-NN method.

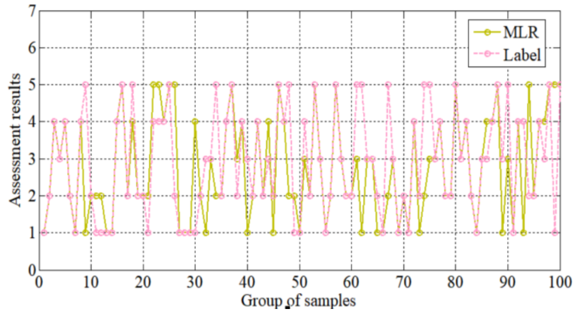

FIGURE 13. The assessment result with the MLR method.

Experimental results show that the proposed method and Bayesian method have the same change trend of the assessment, but the assessment results of different methods are different. The proposed method has better assessment results. The deep learning model shows its advantages in learning a nonlinear relationship. Deep learning provides a good solution for situation assessment. However, the global scene information is fed to the deep learning model, and the evaluation

TABLE 2. The first standardized scene data.

Label	First time of the normalized samples								
F1	-0.5556	-0.6667	-0.7222	-0.8333	-0.6111	-0.3889	-0.6667	-0.6111	-0.9444
F2	0.2500	-0.1667	0.1667	-0.0833	0.3333	0.5833	-0.0833	0.1667	-0.2500
F3	-0.8644	-0.8644	-0.7966	-0.8305	-0.8644	-0.7627	-0.8305	-0.8305	-0.8644
F4	0.5833	-0.9167	0.1667	-0.9167	0.5833	-0.7500	-1.0000	0.6667	0.2500

TABLE 3. The second standardized scene data.

Label	Second time of the normalized samples								
F1	-0.1111	-0.0556	0.2778	-0.1667	0.8889	0.1667	-0.0556	0.4444	0.0556
F2	-0.1667	-0.8333	-0.1667	-0.4167	-0.3333	-0.3333	-0.1667	-0.0833	0.1667
F3	0.0847	0.0169	0.1864	0.0508	0.9322	0.5593	0.2881	0.4915	0.4915
F4	0.1667	-0.2500	0.1667	-0.2500	0.5833	0.7500	0.4167	0.6667	0.8333

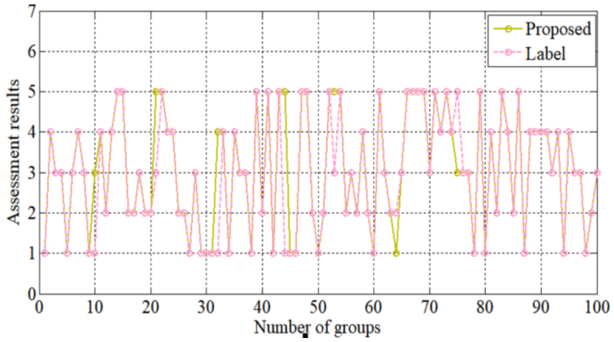

FIGURE 14. The assessment result with the proposed method.

results ignore the differences between agents. We normalize the test samples and use the normalized samples as the input for the proposed method. Different situation assessment models will output different situation assessment results. We recorded every 10 groups. 100 groups of test data were recorded 10 times. Table 2 to Table 3 shows the normalized results for the 18 groups.

Table 2 shows the normalized scene data obtained for the first time. Table 2 shows the data obtained for the second time. The normalization method for the scene data is given by Eq. (3) - Eq. (5).

In Fig.10 and Fig.11, the horizontal axis represents the number of the groups for samples, and the vertical axis represents the assessment results. In the test period, 50 groups of normalized data are input into the two network models, and the final results are compared. The results are shown in Fig. 12 to Fig. 14. From the experimental results, the average accuracy of the proposed deep learning method is about 90%, the BP-NN is about 80% and the MLR method is about 59%. The proposed deep learning-based method has a more accurate assessment result.

The evaluation effect of MLR is the worst, which shows that the sample linear relationship can not describe the mapping from scene factors to situation assessment results. Deep learning outperforms the performance of classical neural networks, and this result is consistent with the previous work. Deep learning can provide a more powerful tool for nonlinear function fitting. Collaborative intelligence brings together the decision-making results of the participants in a way of voting rather than autocracy. This method can get better results.

V. CONCLUSION

In this work, to address the situation assessment for a multi-agent system, a deep learning-based method with Fuzzy fusion is proposed. A deep learning method is developed to achieve local situation assessment. Meanwhile, the information sharing integrating the shortest path faster algorithm is developed. Firstly, each agent receives the state information from others and uses it as the input for the deep learning computing model. A standardization method is proposed to uniform the scene data. Then, the deep learning method running several assessments to get the results for local situation assessment. The average accuracy of the proposed method is more than 90%, which is about 10% - 20% higher than the conventional methods. The experimental results show that the proposed scheme outperforms the competitors in terms of assessment accuracy and efficiency. The deep learning model shows its advantages in fitting the nonlinear situation assessment relationship. A collaborative intelligence method integrating fuzzy logic considers the characteristics of all agents, so it gets better results.

The proposed model uses the deep learning and fuzzy method, which can be generalized to other task scenarios, even other multi-agent systems, via a way of knowledge transfer. In the future, we will extend the proposed method to more multi-agent systems. An adaptive strategy selection method is urgently needed to cooperate with situation assessment to achieve efficient decision-making [32], [33]. Besides, integrating the possibility of applying an advanced deep learning model [34], [35] for situation fusion is also a research direction.

ACKNOWLEDGMENT

(Peng Liao and Mg Xu contributed equally to this work.)

REFERENCES

[1] Z. Li, L. Gao, W. Chen, and Y. Xu, "Distributed adaptive cooperative tracking of uncertain nonlinear fractional-order multi-agent systems," *IEEE/CAA J. Automatica Sinica*, vol. 7, no. 1, pp. 292–300, Jan. 2020, doi: 10.1109/JAS.2019.1911858.

[2] S. Piao and B. Hong, "Fast reinforcement learning approach to cooperative behavior acquisition in multi-agent system," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Lausanne, Switzerland, Sep./Oct. 2002, pp. 871–875, doi: 10.1109/IRDS.2002.1041500.

[3] T. T. Nguyen, N. D. Nguyen, and S. Nahavandi, "Deep reinforcement learning for multiagent systems: A review of challenges, solutions, and applications," *IEEE Trans. Cybern.*, vol. 50, no. 9, pp. 3826–3839, Sep. 2020, doi: 10.1109/TCYB.2020.2977374.

[4] T. Chu, J. Wang, L. Codeca, and Z. Li, "Multi-agent deep reinforcement learning for large-scale traffic signal control," *IEEE Trans. Intell. Transp. Syst.*, vol. 21, no. 3, pp. 1086–1095, Mar. 2020, doi: 10.1109/TITS.2019.2901791.

[5] D. Cao, W. Hu, J. Zhao, Q. Huang, Z. Chen, and F. Blaabjerg, "A multi-agent deep reinforcement learning based voltage regulation using coordinated PV inverters," *IEEE Trans. Power Syst.*, vol. 35, no. 5, pp. 4120–4123, Sep. 2020, doi: 10.1109/TPWRS.2020.3000652.

[6] Y. Ye, D. Qiu, J. Li, and G. Strbac, "Multi-period and multi-spatial equilibrium analysis in imperfect electricity markets: A novel multi-agent deep reinforcement learning approach," *IEEE Access*, vol. 7, pp. 130515–130529, 2019, doi: 10.1109/ACCESS.2019.2940005.

[7] P. Shi and Q. Shen, "Cooperative control of multi-agent systems with unknown state-dependent controlling effects," *IEEE Trans. Autom. Sci. Eng.*, vol. 12, no. 3, pp. 827–834, Jul. 2015, doi: 10.1109/TASE.2015.2403261.

[8] Y. Liu and H. Lu, "A strategy of multi-UAV cooperative path planning based on CCPSO," in *Proc. IEEE Int. Conf. Unmanned Syst. (ICUS)*, Beijing, China, Oct. 2019, pp. 328–333, doi: 10.1109/ICUS48101.2019.8995982.

[9] J. Kim, K. Jo, W. Lim, M. Lee, and M. Sunwoo, "Curvilinear-coordinate-based object and situation assessment for highly automated vehicles," *IEEE Trans. Intell. Transp. Syst.*, vol. 16, no. 3, pp. 1559–1575, Jun. 2015, doi: 10.1109/TITS.2014.2369737.

[10] G. P. Settembre, D. Nardi, R. Pigliacampo, A. Farinelli, and M. Rossi, "Solving disagreements in a multi-agent system performing situation assessment," in *Proc. 12th Int. Conf. Inf. Fusion*, Seattle, WA, USA, 2009, pp. 717–724.

[11] G. Jakobson, "Using federated adaptable multi-agent systems in achieving cyber attack tolerant missions," in *Proc. IEEE Int. Multi-Disciplinary Conf. Cognit. Methods Situation Awareness Decis. Support*, New Orleans, LA, USA, Mar. 2012, pp. 96–102, doi: 10.1109/CogSIMA.2012.6188415.

[12] L. Fu, J. Liu, F. Chang, and G. Meng, "Situation assessment for beyond-visual-range air combat situation assessment based on dynamic Bayesian network," in *Proc. 25th Chin. Control Decis. Conf. (CCDC)*, Guiyang, China, May 2013, pp. 591–594, doi: 10.1109/CCDC.2013.6560993.

[13] L. Zihao, Z. Bin, Z. Ning, and L. Lixun, "Hierarchical network threat situation assessment method for DDoS based on D-S evidence theory," in *Proc. IEEE Int. Conf. Intell. Secur. Informat. (ISI)*, Beijing, China, Jul. 2017, pp. 49–53, doi: 10.1109/ISI.2017.8004873.

[14] J. Zhao and S. Lu, "The assessment of environment effect of foreign trade based on BP neural network and multipe linear regression," in *Proc. Asia–Pacific Conf. Comput. Intell. Ind. Appl. (PACIIA)*, Wuhan, China, Nov. 2009, pp. 300–303, doi: 10.1109/PACIIA.2009.5406431.

[15] T. Fu, Y. Lu, and W. Zhen, "APT attack situation assessment model based on optimized BP neural network," in *Proc. IEEE 3rd Inf. Technol., Netw., Electron. Autom. Control Conf. (ITNEC)*, Chengdu, China, Mar. 2019, pp. 2108–2111, doi: 10.1109/ITNEC.2019.8729178.

[16] M. Li, H. Cao, X. Song, Y. Huang, J. Wang, and Z. Huang, "Shared control driver assistance system based on driving intention and situation assessment," *IEEE Trans. Ind. Informat.*, vol. 14, no. 11, pp. 4982–4994, Nov. 2018, doi: 10.1109/TII.2018.2865105.

[17] G. Ruiz-Garcia, H. Hagras, H. Pomares, and I. R. Ruiz, "Toward a fuzzy logic system based on general forms of interval type-2 fuzzy sets," *IEEE Trans. Fuzzy Syst.*, vol. 27, no. 12, pp. 2381–2395, Dec. 2019, doi: 10.1109/TFUZZ.2019.2898582.

[18] X. U. Ximeng, Y. Rennong, and F. U. Ying, "Situation assessment for air combat based on novel semi-supervised naive Bayes," *J. Syst. Eng. Electron.*, vol. 29, no. 4, pp. 768–779, 2018.

[19] A. Yang, Y. Zhuansun, C. Liu, J. Li, and C. Zhang, "Design of intrusion detection system for Internet of Things based on improved BP neural network," *IEEE Access*, vol. 7, pp. 106043–106052, 2019, doi: 10.1109/ACCESS.2019.2929919.

[20] H. Zhu, N. Lin, H. Leung, R. Leung, and S. Theodoidis, "Target classification from SAR imagery based on the pixel grayscale decline by graph convolutional neural network," *IEEE Sensors Lett.*, vol. 4, no. 6, pp. 1–4, Jun. 2020, doi: 10.1109/LSENS.2020.2995060.

[21] P. Zhang, X. Wang, W. Zhang, and J. Chen, "Learning spatial–spectral–temporal EEG features with recurrent 3D convolutional neural networks for cross-task mental workload assessment," *IEEE Trans. Neural Syst. Rehabil. Eng.*, vol. 27, no. 1, pp. 31–42, Jan. 2019, doi: 10.1109/TNSRE.2018.2884641.

[22] N. Passalis and A. Tefas, "Training lightweight deep convolutional neural networks using bag-of-features pooling," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 30, no. 6, pp. 1705–1715, Jun. 2019, doi: 10.1109/TNNLS.2018.2872995.

[23] J. Garland and D. Gregg, "Low complexity multiply accumulate unit for weight-sharing convolutional neural networks," *IEEE Comput. Archit. Lett.*, vol. 16, no. 2, pp. 132–135, Jul. 2017, doi: 10.1109/LCA.2017.2656880.

[24] J. Lim, M. Potsdam, R. Strawn, B. Sim, and T. Nygaard, "Blade-vortex interaction airloads prediction using multidisciplinary coupling," in *Proc. HPCMP Users Group Conf. (HPCMP-UGC)*, Denver, CO, USA, 2006, pp. 59–65, doi: 10.1109/HPCMP-UGC.2006.7.

[25] L. Ruan, J. Wang, J. Chen, Y. Xu, Y. Yang, H. Jiang, Y. Zhang, and Y. Xu, "Energy-efficient multi-UAV coverage deployment in UAV networks: A game-theoretic framework," *China Commun.*, vol. 15, no. 10, pp. 194–209, Oct. 2018, doi: 10.1109/CC.2018.8485481.

[26] Q. Qinqin, G. Chao, Q. Lei, and L. Yue, "Congestion control strategy for power scale-free communication network by link addition," in *Proc. 14th IEEE Int. Conf. Electron. Meas. Instrum. (ICEMI)*, Changsha, China, Nov. 2019, pp. 1714–1722, doi: 10.1109/ICEMI46757.2019.9101412.

[27] R. A. Rabelo, H. T. Macedo, and E. O. Freire, "The SimuroSot strategy development kit: A high-level approach to robot soccer coding," *IEEE Latin Amer. Trans.*, vol. 16, no. 2, pp. 686–693, Feb. 2018, doi: 10.1109/TLA.2018.8327430.

[28] P. Wawrzynski, P. Zawistowski, and L. Lepak, "Automatic hyperparameter tuning in on-line learning: Classic momentum and ADAM," in *Proc. Int. Joint Conf. Neural Netw. (IJCNN)*, Glasgow, U.K., Jul. 2020, pp. 1–8, doi: 10.1109/IJCNN48605.2020.9207204.

[29] R. A. Suarez-Rivera, A. Loza-Alvarez, J. R. Espinosa-Baylon, S. Martinez-Cruz, K. Perez-Alvarez, D. Jaramillo-Arteaga, G. I. Perez-Soto, and K. A. Camarillo-Gomez, "Adaptation of robot BIOLOID premium for weightlifting HuroCup—FIRA," in *Proc. 18th Congreso Mexicano de Robotica*, Sinaloa de Leyva, Mexico, Nov. 2016, pp. 1–3, doi: 10.1109/COMROB.2016.7955156.

[30] H. Zhang, L. Chen, H. Wang, Y. Zhao, L. Chen, and B. Duan, "Island partition of distribution network with microgrid and distributed generation based on floyd-warshall algorithm," in *Proc. Int. Conf. Smart Grid Electr. Autom. (ICSGEA)*, Changsha, China, Jun. 2018, pp. 29–33, doi: 10.1109/ICSGEA.2018.00015.

[31] M. C. Silva, A. C. B. Garcia, and A. Conci, "A multi-agent system for dynamic path planning," in *Proc. 2nd Brazilian Workshop Social Simulation*, Sao Bernardo do Campo, São Paulo, Oct. 2010, pp. 47–51, doi: 10.1109/BWSS.2010.20.

[32] M. Zhao, H. Lu, S. Yang, and F. Guo, "The experience-memory Q-learning algorithm for robot path planning in unknown environment," *IEEE Access*, vol. 8, pp. 47824–47844, 2020, doi: 10.1109/ACCESS.2020.2978077.

[33] C. Hu and M. Xu, "Adaptive exploration strategy with multi-attribute decision-making for reinforcement learning," *IEEE Access*, vol. 8, pp. 32353–32364, 2020, doi: 10.1109/ACCESS.2020.2973169.

[34] S. Kuanar, K. R. Rao, D. Mahapatra, and M. Bilas, "Night time haze and glow removal using deep dilated convolutional network," 2019, *arXiv:1902.00855*. [Online]. Available: http://arxiv.org/abs/1902.00855

[35] S. Kuanar, K. R. Rao, M. Bilas, and J. Bredow, "Adaptive CU mode selection in HEVC intra prediction: A deep learning approach," *Circuits, Syst., Signal Process.*, vol. 38, no. 11, pp. 5081–5102, Nov. 2019.

PENG LIAO was born in Anhui Province, China, in 1996. He received the B.S. degree in aircraft design and engineering and the M.S. degree in flight vehicle design from Northwestern Polytechnical University, Xi'an, China. He is currently an Assistant Engineer with the China Academy of Aerospace Aerodynamics. His research interest includes artificial intelligence in aircraft design.

MG XU is currently an Engineer with the Software Research and Development Department, Xi'an Bazhentu Network Technical Cooperation. His research interests include deep learning, machine learning, and software engineering.

CONGYING YANG is currently a Division Director with the Software Research and Development Department, Xi'an Bazhentu Network Technical Cooperation. Her research interests include machine learning and software engineering.