IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

SPECIAL SECTION ON DEEP LEARNING FOR INTERNET OF THINGS

Received August 11, 2021, accepted September 1, 2021, date of publication September 3, 2021,
date of current version September 17, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3110342

The Direction Analysis on Trajectory of Fast Neural Network Learning Robot

XIAOHONG LI[1,2] AND MAOLIN LI[1]

[1]School of Information Engineering, Shaoyang University, Shaoyang, Hunan 422000, China
[2]Graduate School, Adamson University, Manila 1000, Philippines

Corresponding author: Xiaohong Li (lisyad80@163.com)

This work was supported in part by the General Project of the Department of Education of Hunan Province (Research on the Prediction Method of College Students' Achievement Based on Machine Learning) under Grant 20C1656, in part by the General Project of Shaoyang Science and Technology Bureau (Research on the Application of Collaborative Filtering Recommendation Algorithm in Personalized Agricultural Information Recommendation Service) under Grant 2018NS26, and in part by the Subject of Educational Science Planning in Hunan Province (Research on the Teaching Innovation of Programming Course in Colleges and Universities Based on the Demand of Deep Learning) under Grant ND206628.

ABSTRACT This study is to efficiently apply artificial neural network (ANN) to the robotics, so as to provide experimental basis for mobile robots to learn the optimal trajectory planning strategy. An algorithm model is innovatively proposed based on back propagation neural network (BPNN) and reinforcement learning (Q-Learning) by combining the motion space, selective strategy, and reward function design. The simulation experiment environment is set and the ROS mobile robot is adopted for simulation experiments. The algorithm proposed in this study is compared with other neural network algorithms from the perspectives of accuracy, precision, recall, and F1. It can be found that the accuracy of algorithm proposed was at least 5.47% higher than that of the model algorithm proposed by other scholars, and the values of precision, recall, and F1 were at least 5.5% higher. The results show that the mobile robot could find the shortest trajectory and the best trajectory in a discrete obstacle environment, no matter the more or less the discrete obstacles or the large or small the space. Therefore, compared to the advanced model algorithms proposed by other scholars in related fields, the robot trajectory planning based on the improved BPNN combined with Q-Learning constructed in this study could realize better results, and can be used in practical applications with robot trajectory planning, providing practical value for the field of machine vision.

INDEX TERMS Artificial neural network, back propagation neural network, Q-learning, mobile robot.

I. INTRODUCTION

Many disciplines have made great progress under rapid development of science and technology today. Driven by academic and industrial needs, the development of multidisciplinary technology integration is very fast, and many excellent results have been obtained in the academic world and in the process of production practice [1], [2]. Robot-related technology is a representative of multidisciplinary technology integration. It integrates the research results of computers, sensors, and artificial intelligence (AI). It is the pinnacle of mechatronics achievements and can represent the high-tech level of a country [3]. The Stanford Research Institute firstly began to study the autonomous path planning capabilities of autonomous mobile robots in complex environments in the 1960s [4]. At present, an important development direction for various countries in the world is related to the research and development of robots. China has clearly pointed out in its future development plan that robots, especially robots with autonomous mobility, will be included in the field of advanced manufacturing technology [5], [6]. Germany pointed out in 2013 that the future development of frontier industries should give priority to the combination of intelligent robots and man-machines. In 2018, Japan also proposed to include the smart manufacturing and mobile robots in the five key development areas. In the United States, it was also clearly pointed out that the research on robots and autonomous systems would be included in the next 20 technological trends. This means that in the future, robotics is the main direction of scientific and technological competition among countries in the world.

The associate editor coordinating the review of this manuscript and approving it for publication was Chi-Hua Chen.

The movement trajectory planning of mobile robots is the core research point during the research. The trajectory planning of a mobile robot is mainly based on what kind of trajectory the robot walks on with or without a map [7]. The local movement trajectory planning of a mobile robot is a dynamic planning method, and its most important feature is that the mobile robot can realize the real-time movement trajectory planning based on local environment information. However, the environment for local movement trajectory planning of mobile robots is unpredictable, and it is difficult to deal with various situations through experience. Therefore, it is necessary to introduce a self-learning function, so that the robot can navigate autonomously and avoid obstacles after a period of training [8], [9]. The reinforcement learning algorithm has been widely used in the local trajectory planning of mobile robots, and it has been proved through practice to be an effective algorithm for improving the intelligent system, including the clustering algorithm [10]. Reinforcement learning algorithm takes the ''trial and error'' behavior as the basis, it uses the delayed return method to find the optimal action to obtain the best decision-making ability. The core feature of reinforcement learning is that it can learn online and update itself, which is one of the core technologies of path planning. Reinforcement learning has become more and more mature in algorithm theory and application by combining algorithms and disciplines such as neural networks, intelligent control, and game theory [11], [12]. Q-Learning is the most commonly used reinforcement learning algorithm, but its convergence rate is relatively low [13]. The back propagation neural network (BPNN) under artificial neural network (ANN) shows excellent perception and computing capabilities, is good at nonlinear prediction and fitting, and can adjust the connection signal strength among neurons to learn external environmental knowledge, showing a strong generalization ability. Therefore, BPNN has become an important calculation model for mobile robot motion behavior control [14]. Some researchers have combined the potential field method with ANN to improve the effect of movement trajectory planning of mobile robot in a dynamic environment [15], but it can't solve a series of problems such as the slow convergence speed of ANN represented by BPNN.

Based on above contents, the additional momentum method is combined with the adaptive learning rate method to optimize BPNN, and a trajectory planning based on the BPNN and Q-Learning is innovatively proposed, so as to provide effective experimental basis for the robots to learn the best trajectory planning strategy under various obstacles conditions and for the better development of the robotics.

II. PREVIOUS WORKS

A. ANALYSIS ON ANN

With the rapid development of science and technology, deep learning has been extensively studied in various fields. As the key content of deep learning, ANN currently also occupies an important position in the field of robotics. Al-Qurashi and Ziebart [16] studied the use of Long Short Term Memory - Recurrent Neural Networks (LSTM-RNN) to optimize the trajectory of the robot, which is superior to other neural networks in position and direction. Peng *et al.* [17] proposed the use of Radial Basis Function Network (RBF network) to solve the uncertainty of the robot control model, and verified the effectiveness of the method.

B. ANALYSIS ON ROBOT MOVEMENT TRAJECTORY

In recent years, robots have attracted more and more attention from researchers in the form of human-like or animal-like robots. Prasetyo *et al.* [18] researched and discussed the gait planning proposed on the quadruped robot, the trajectory planning used under this case is linear translation and sinusoidal gait trajectory, and there are no obstacles, just walking on flat terrain. Liu and Wang [19] proposed a local trajectory planning method for ground service robots, which generates a feasible and comfortable trajectory while considering multiple stationary obstacles and path curvature constraints.

C. LITERATURE REVIEW

Robot trajectory planning faces some delay in avoiding obstacles, inaccurate path planning, and low model optimization efficiency. The existing ANN is very weak when used in robot trajectory planning, so it is unable to accurately obtain the data set, and unable to accurately establish a mathematical model suitable for robot trajectory planning. In view of the above shortcomings, most of the current researches use a single artificial neural network for robot trajectory planning. However, the algorithms for robot trajectory planning is simple due to the lack of complete data sets and limited model training capabilities, so there are few researchers optimize and recombine the neural network algorithms.

III. THE TRAJECTORY PLANNING ALGORITHM BASED ON THE BPNN AND Q-LEARNING

A. Q-LEARNING

The mobile robot can optimize the task result by continuously interaction with the environment. When interacting with the environment with a certain action, the mobile robot will generate a new state under the action of the motion and the environment, which will be rewarded immediately by the environment. During the constant repetition, the mobile robot continuously interacts with the environment, generating a large amount of data. The Q-Learning algorithm optimizes its own action strategy through the generated data, and then interacts with the environment to generate new data, which can be adopted to further improve its own mobile strategy. After many iterations of learning, the mobile robot finally learns to obtain the best action sequence to complete the corresponding task. Therefore, the theoretical theories of reinforcement learning are learnt firstly in this study. Q-Learning is a reward guidance behavior obtained by the agent through ''trial and error'' learning and interaction with the environment, aiming to maximize the reward for the

agent. Q-Learning is different from supervised learning in connectionist learning, which is mainly reflected in the reinforcement signal. The reinforcement signal provided by the environment in reinforcement learning is used to judge the quality of the action. It does not tell the Q-Learning system how to generate the correct action, but an evaluation, usually a scalar signal. Because the external environment provides little information, reinforcement learning system (RSL) must learn from one's own experience. In this way, RLS gains knowledge in the action evaluation environment and improves the action plan to adapt to the environment.

Q-Learning [20] is a model-independent reinforcement learning algorithm, which can directly optimize a Q function that can be calculated iteratively. Its target strategy is greedy strategy, and its action strategy is ε-greedy. The algorithm steps of Q-Learning are shown in Figure 1 below:

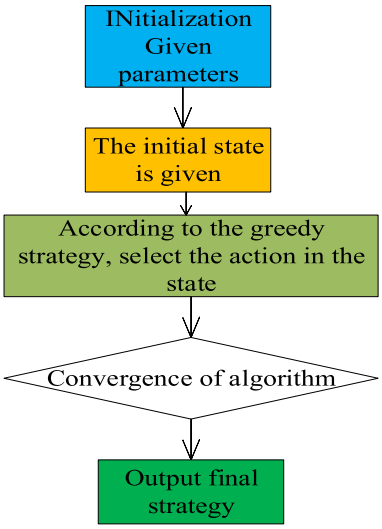

FIGURE 1. The algorithm steps of Q-learning.

After the algorithm is initialized and the parameters are set, the actions are selected according to the strategy in the initial state, the rewards and the next state are received, the end state is realized after convergence, and the final strategy is outputted.

B. Q VALUE FUNCTION PREDICTION MODEL BASED ON ANN

During the Q-Learning, mobile robots show slow convergence speed, but ANN shows very fast perception calculation speed and good nonlinear prediction and fitting, can adjust the connection signal strength among neurons, and can learn about the external environment. Therefore, the movement trajectory of the mobile robot is optimized based on the ANN.

A neural network is a complex non-linear network composed of a large number of simple non-linear units. It is a non-linear model to simulate the function of the human brain. Essentially, it is a model-independent adaptive function estimator. When the given input is not the original training sample, the neural network can also give an appropriate output, that is, it has generalization ability. In the neural network, knowledge is distributed in the storage network through learning examples, so the neural network is fault-tolerant. When a single processing unit is damaged, it has little effect on the overall behavior of the neural network, but it does not affect the normal operation of the entire system. Because of its strong learning ability and nonlinear mapping ability, neural network has been widely used in robot kinematics, dynamics, and control. In mobile robot navigation, it is mainly used for environment model representation, local planning, global planning, sensor information fusion, robot control system, etc. The ANN is a nonlinear adaptive information processing system composed of a large number of interconnected processing units. It is proposed on the basis of the results of modern neuroscience research, trying to process information by simulating the processing and memory of the brain neural network [21]. There are four basic characteristics for ANN, as shown in Table 1 below:

TABLE 1. Basic characteristics of ANN.

Characteristic category	Basic contents
Non-linear	Artificial neurons are in two different states of activation or inhibition, which is a non-linear relationship in mathematics. Neural networks with thresholds have better performance and can improve fault tolerance and storage capacity.
Unrestricted	Neural networks are usually composed of many neurons. The overall behavior of a system depends not only on the characteristics of a single neuron, but also on the interaction and interconnection among various units. The infinite functions of the brain can be simulated through a large number of connections among the units.
Non-constancy	ANN is capable of self-adaptation, self-organization, and self-learning. Not only will the information processed by the neural network undergo various changes, but the nonlinear dynamic system itself is also constantly changing. The iterative process is often used to describe the evolution process of a dynamic system.
Non-convexity	The function has multiple extreme values, so the system has multiple stable equilibrium states, which will lead to the diversity of system evolution.

ANN can realize the abstract analysis on the neural network of the human brain from the perspective of information processing and establishes a simple model, forming different networks using different connection methods. It is a running model composed of many interconnected nodes (neurons). Each node (neuron) represents a specific output function, which is called the activation function [22]. Each connection between two nodes represents a weighted value of the signal passing through the connection, which is called the weight, being equivalent to the memory of the ANN. The output of the network varies with the connection mode, weight, and

excitation function of the network. In essence, the network itself is usually an approximation of a specific algorithm or function, or an expression of a logic strategy. It is assumed that the input weight of neuron n is φ, the activation function is f, and the accumulation unit is m, then the output G of the neuron can be expressed as equation (1):

$$G_{\mathrm{n}} = f\left(\sum_{n} \varphi_n x_n + m\right), \tag{1}$$

The input weight in the above equation acts on the sample output x or sample input of the upper layer of the network, so as to obtain the cumulative result by sum. Then, the nonlinear activation function f is adopted to obtain the response value. The activation function is a threshold analysis mechanism, which can be activated and output only when the input exceeds a certain value, thus forming a neuron in the neural network [23]. The structure of the neuron activation function is shown in Figure 2 below:

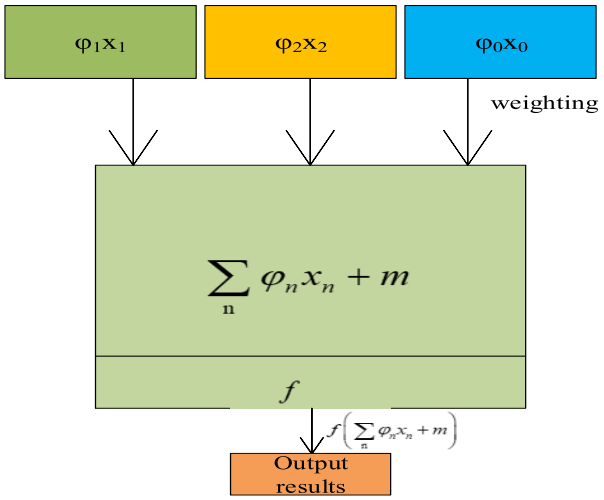

FIGURE 2. The structure diagram of the neuron activation function.

A neural network is a complex non-linear network composed of a large number of simple non-linear units. Essentially, it is a model-independent adaptive function estimator. When the given input is not the original training sample, the neural network can also give an appropriate output, that is, it has generalization ability. In the neural network, knowledge is distributed in the storage network through learning examples, so the neural network is fault-tolerant. When a single processing unit is damaged, it has little effect on the overall behavior of the neural network, but it does not affect the normal operation of the entire system. Because of its strong learning ability and nonlinear mapping ability, neural network has been widely used in the robot kinematics, dynamics, and control. In mobile robot navigation, it is mainly used for environment model representation, local planning, global planning, sensor information fusion, robot control system, etc. The ANN is a nonlinear adaptive information processing system composed of a large number of interconnected processing units. It is proposed on the basis of the results of modern neuroscience research, trying to process information by simulating the processing and memory of the brain neural network [24].

The back propagation neural network (BPNN) is a type of ANN. The basic BPNN algorithm is composed of the forward propagation of signals and backward propagation of errors. In other words, the error output is calculated according to the direction of input to output, and the weight and threshold are adjusted according to the direction of output to input. In the forward propagation, the input signal acts on the output node through the hidden layer, and the output signal is generated through nonlinear transformation. If the actual output is inconsistent with the expected output, it will be transformed into an error back propagation process. Error retransmission is to retransmit output errors to the input layer through the hidden layer and distribute the errors to all units in each layer. The error signal of each layer is undertaken as the basis for adjusting the weight of each unit. The error is reduced along the gradient direction through adjusting the connection strength between the input node and the hidden node, the connection strength between the hidden node, and the output node and the threshold. After repeated learning and training, the network parameters (weights and thresholds) corresponding to the minimum error are determined, and the training is stopped. At this time, the trained neural network can process the input information of similar samples on its own, as well as the nonlinear conversion information with the smallest output error [25], [26]. BPNN is capable of non-linear mapping and can approximate any continuous function. Aiming at this mapping ability, the collected state-actions are used to evaluate the Q-value function, and BPNN training is performed at the same time. Finally, a Q-value function prediction model is analyzed from the multiple state-action data obtained by the intensive Q-Learning, and the model is applied for prediction of new Q value data. BPNN shows the disadvantages of long learning time, slow convergence, and network training easily falling into local minimums. In view of the above shortcomings, the additional momentum method is combined with the adaptive learning rate method [27] in this study to apply the advantages of them to improve the BPNN algorithm. The learning rate of the network is set to δ. If the deviation of the system feedback is gradually reduced, the next learning rate will increase, and vice versa, the learning rate will decrease. If the network system is trained to the saturation area of the error surface, the variation of the error is very small at this time, then the additional momentum term method can be expressed as equation (2) below:

$$\Delta\mu \approx \frac{\alpha}{1-\lambda}\frac{\kappa Q}{\kappa\mu}, \tag{2}$$

In the above equation, α represents the learning rate, λ represents the momentum factor in the network system, Q is the network error, and κ refers to the allowable rebound error coefficient.

At this time, the adjustment process of the system network connection weight can be expressed as follows:

$$\Delta\mu\,(t+1) = \alpha\frac{\kappa Q}{\kappa R} + \lambda\Delta\mu\,(t)\,, \tag{3}$$

In the equation above, t represents time. The adjustment of online learning rate through the unevenness of the deviation surface can improve the convergence of the BPNN algorithm and effectively solve the drastic changes in the error curve.

In ANN, RNN and LSTM are also often used in robot trajectory planning. The difference among RNN, LSTM, and BPNN is that LSTM can only avoid the disappearance of the gradient of RNN, but it can't combat the gradient explosion, while BPNN shows strong self-learning, adaptive capabilities, generalization capabilities, nonlinear mapping capabilities, and fault tolerance, so it is more suitable for the computer vision.

First of all, the prediction model of the BPNN combined with Q-value function designed in this study is also composed of an input layer, a hidden layer, and an output layer, but the hidden layer here is structured with a single layer. The input of the neural network model is the environmental state variables A, B, C, and D perceived by the mobile robot, then the state vector dimension u of the input layer is 5, and the output layer v is the 4 Q values corresponding to each action: $Q(S_t,x_1)$, $Q(S_t,x_2)$, $Q(S_t,x_3)$, and $Q(S_t,x_4)$. The number of neurons in the hidden layer can be calculated according to the number of neurons in the input layer and output layer, as follows:

$$N \le \sqrt{(u+v)} + \mathrm{k}, \tag{4}$$

In the above equation (2), u is the dimension of the state vector of the input layer, v refers to the output layer, k represents a constant within [1], [10]. Therefore, it can be known that the number of neurons in the hidden layer is in the range of [4], [13]. In order to maximize the training effect of the model, the number of neurons is selected as 13 in this study.

The structure of the Q value prediction model based on the BPNN is given as follows (Figure 3).

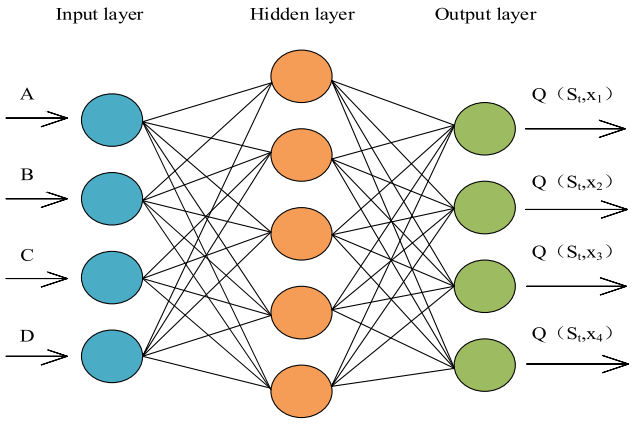

FIGURE 3. The structure of the Q value prediction model based on the BPNN.

The input layer in the above model inputs the environmental variables perceived by the robot, the output layer outputs the Q value of each action of the robot through the feature conversion of the hidden layer, and then the number of neurons in the hidden layer can be calculated.

The neural network layer contains many neurons, and each neuron is related to each other through weighting, forming an interconnected neural network structure. The most basic ANN consists of an input layer, a hidden layer, and an output layer [28]. The functional characteristics of each layer are shown in Table 2 below:

TABLE 2. The functional characteristics of each layer of the ANN.

Layer name	Basic functions
Input layer	It only receives information from the external environment. The external environment is composed of input units that can receive various types of characteristic information in the sample. Each neuron in this layer is equivalent to an independent variable, which only transmit the information to the next layer without any calculation.
Hidden layer	Located between the input layer and the output layer, the hidden layer is to perform the data weighting calculation and to link the input layer and the output layer through the function. Its calculation result is the input value of the output layer.
Output layer	The output layer is to obtain the calculation result, and each output unit corresponds to a specific classification or a predicted value.

The linear function purelin is undertaken as the training function of the neurons in the output layer, and the S-type differentiable tangent function tansig is selected as the training function of the hidden layer neurons of the BPNN-based Q value prediction model, as shown in the following equation:

$$f\,(x) = \frac{2}{1+e^{-2x}} - 1, \tag{5}$$

The backpropagation error of the BPNN is supposed as e_n, then the error is the actual Q value ($Q(S_t, X_n)$) obtained by the Q learning algorithm minus the Q value ($\widetilde{Q}\,(S_t, X_n)$) predicted by the BPNN under the same set of samples, as shown in the following equation (4):

$$e_n = Q\,(S_t, X_n) - \widetilde{Q}\,(S_t, X_n)\,, \quad \mathrm{n} = 1, 2, 3 \tag{6}$$

The training steps of the BPNN-based Q neural network are given in Table 3.

C. DESIGNS OF MOTION SPACE, SELECTIVE STRATEGY, AND REWARD FUNCTION OF ROBOTS

First, four actions are designed in this study, including forward R_1, back R_2, left turn R_3, and right turn R_4 to allow the mobile robot to walk freely in the map environment. The above actions constitute the action set $R = \{R_1, R_2, R_3, R_4\}$.

TABLE 3. The training steps of the BPNN-based Q neural network.

Steps	Specific contents
Step 1	Acquisition of input data: the mobile robot uses the Q-Learning algorithm to obtain the perception feature vector in the map environment, and normalizes the data in the vector into the input value of the BPNN combined with the Q value prediction model. According to the state behavior relationship of the robot and the obstacle avoidance rule after vibration, the movement behavior of the mobile robot is restricted, and the current state behavior after the robot performs a step is given to evaluate the actual Q value. The feature quantity and expected Q value are saved, and a sample training data is added to the input data set.
Step 2	Adjustment of the parameters of BPNN: the training data is input into the input layer of BPNN, and the weights and thresholds among the three layers are adjusted to minimize the error between the expected value of Q and the predicted value.
Step 3	Judgment of the convergence of BPNN: the sum of squares of multiple inspection errors is undertaken as the evaluation function; and the BPNN converges when the value of the evaluation function is smaller than the given precise value.
Step 4	The weights and thresholds of the BPNN connection layer are saved.

The left and right turning angles are set to be 30° and can be adjusted according to actual needs during operation, and the back movement stipulates that the mobile robot rotates 180° in place before moving forward.

Secondly, a mobile robot sensor model is designed based on the widely used ROS robot sensors [29], including three sonar sensors on the left, center, and right in front of the ROS robot, and the angle between each sensor is 22.5°. The position of the ROS robot body is supposed as the coordinate origin (0,0), the front direction of the robot is the y-axis, and the direction perpendicular to the y-axis is set as the x-axis, and a two-dimensional planar rectangular coordinate system is established as shown in the Figure 4:

In the above coordinate system, the range of the area that the robot can perceive and detect is 30° - 150°. The rectangular plane area is divided into three parts: V_1, V_2, and V_3. Then the left, middle, and right sensors of the robot detect the obstacles in V_1, V_2, and V_3, respectively.

If the mobile robot selects a forward motion r_1 from the motion space R, it will travel to the divided area V_2 of the robot coordinate system in Figure 4. If the left turn action r_2 is selected, it will travel to the area V_3 of the robot coordinate system. If the right turn action r_3 is selected, it will travel to the V_1 area of the robot coordinate system; and if the action R_4 is selected, the robot will move back.

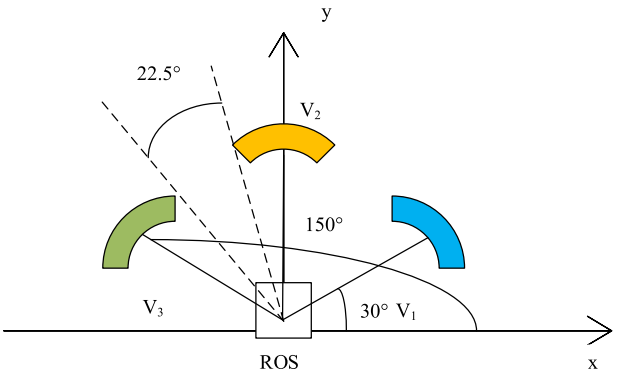

FIGURE 4. Schematic diagram of robot sensor.

Finally, the reward function is designed. The design criterion of the reward function is given as follows. The farther away the obstacle is, the greater the positive reward will be; and the closer the obstacle is, the greater the negative reward. The movement behavior of the mobile robot will be continuously evaluated. Suppose the maximum distance that the ROS robot can perceive is H, the distance between the sensor and the obstacle is h, and the distance relation function between the obstacle and the robot is the logarithmic function of the base number h and $h+0.01$ as the true number, when $h = 0$, the obtained logarithmic function value is not infinite. Then the distance relation function U is expressed as equation (5) below:

$$U(h) = \log_H (h + 0.01), \tag{7}$$

The reward function E of robot to avoiding the obstacles can be written as follows.

$$E = \begin{cases} 0 & \mathrm{h} \geq \mathrm{H} \\ -|U_{t+1}(h) - U_t(h)| & \mathrm{h} < \mathrm{H} \text{ and } U_{t+1}(h) < U_t(h) \\ |U_{t+1}(h) - U_t(h)| & \mathrm{h} < \mathrm{H} \text{ and } U_{t+1}(h) \geq U_t(h) \end{cases}, \tag{8}$$

In the above equation, $U_{t+1}(h)$ and $U_t(h)$ represent the reward values of the distance at time t and $t+1$, respectively.

It is supposed that distance function between the mobile robot and the target is H', which can be expressed as follows:

$$H' = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}, \tag{9}$$

In the equation (7) above, x_1 represents the coordinates of the robot on the x axis, x_2 represents the coordinates of the target on the x axis, y_1 represents the coordinates of the robot on the y axis, and y_2 represents the coordinates of the target on the y axis. According to the above equation, the reward function of the robot approaching the target is designed in the form of a discrete piecewise function, which is expressed

as follows:

$$E' = \begin{cases} \left|H'(t+1) - H'(t)\right| & H'(t+1) < H'(t) \\ 0 & H'(t+1) = H'(t) \\ -\left|H'(t+1) - H'(t)\right| & H'(t+1) > H'(t) \end{cases}, \tag{10}$$

In the equation above, $H'(t+1)$ and $H'(t)$ represent the distance between the robot and the target at $t+1$ and t, respectively. Therefore, the total reward function E_0 is calculated with below equation (9):

$$E_0 = \frac{E + E'}{2}, \tag{11}$$

The smaller the calculation value obtained by the above calculation method, the better beneficial to increase the calculation speed.

D. THE ALGORITHM FLOW OF LOCAL PATH PLANNING BASED ON BPNN-Q LEARNING

The mobile robot obtains state variables according to the environmental state information sensed by the sensor itself, then selects actions, and gets the Q value function table of the convergence function. The optimal state is used as the sample data of the BPNN to obtain the BPNN-Q with generalization ability. The predictive model of the value function. The model predicts the Q value selection action based on the sensor information of the mobile robot on the given unknown environment map, and realizes the local trajectory planning of the mobile robot. The local trajectory planning algorithm of BPNN combined with Q-Learning is shown in Figure 5 below:

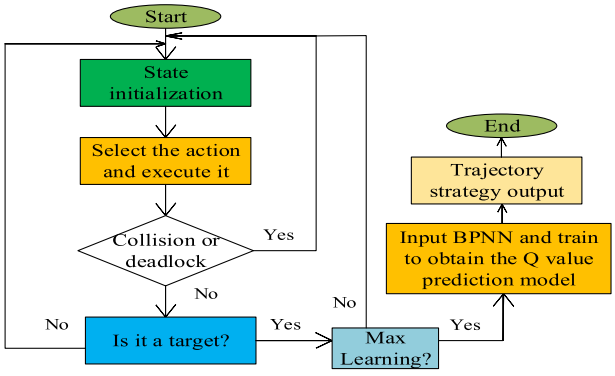

FIGURE 5. The local certain trajectory planning algorithm of BPNN combined with Q-learning.

After the initialization operation (Q table, environment information, and BPNN parameter symbol), the motion strategy of the robot is selected to obtain the result of the reward function and the next state. After the action is executed, collision avoidance detection is performed. If there is a collision, the robot will return to the previous step to re-adjust; if there is no collision, it will get the obstacle avoidance reward, and then check whether it is approaching the target. If it is approaching the target, the target reward will be performed and the total return is calculated; if it is far away from the target, the total return is directly calculated, the Q value function is updated, and then the state is saved. If the current state position is the best, the best sample data is inputted in the BPNN, the parameters are adjusted to obtain the maximum number of iterations, the BPNN-Q value function prediction is obtained, and the best strategy is generated; if it is not the best target position, it returns to the initialization state to recalculate.

E. EXPERIMENTAL ENVIRONMENT AND PARAMETERS SETTING OF THE SIMULATION

In this study, the python's keras library is adopted to train the BPNN, and the Robomaster2019 data set is applied for robot trajectory training.

The simulation experiment is performed on the ROS mobile robot [30], and the hardware and software components of which are shown in Table 4 below:

TABLE 4. The hardware and software components of the simulation.

Experimental environment	Attribute
Hardware	windows 10 operating system (64 bits), 8GB running memory, Intel(R) Core(TM) i5-2450 2.5 GHz central processing unit (CPU)
Software	MATLAB simulation software

The simulation environment is a two-dimensional coordinate, the size of the map is 50×50, and the mobile robot can move freely in the barrier-free area with randomly given steps and directions. The starting point, ending point, and obstacles are randomly set in the environment map.

The parameter settings are shown in Table 5 below:

TABLE 5. The parameter settings of the simulation.

Parameter	Value
Movement step length of the robot	1 cm
Movement radius of the robot	0.5 cm
Movement velocity of the robot	1 cm/s
Minimum and maximum measuring distance of the sensor	1 cm and 5 cm
Learning rate of the algorithm	0.3
Maximum learning times of the robot	2,800 times

IV. RESULTS AND DISCUSSION

A. COMPARISON ON PREDICTION PERFORMANCES OF DIFFERENT ALGORITHMS

To study the performance of the robot trajectory planning based on the improved BPNN combined with Q-Learning

constructed in this study, the algorithm and DDPG combined with ML algorithm, BPNN, DQN combined with ML algorithm, RNN and LSTM are analyzed in terms of accuracy, precision, recall, and the F1 value, and the results are shown in Figure 6 below:

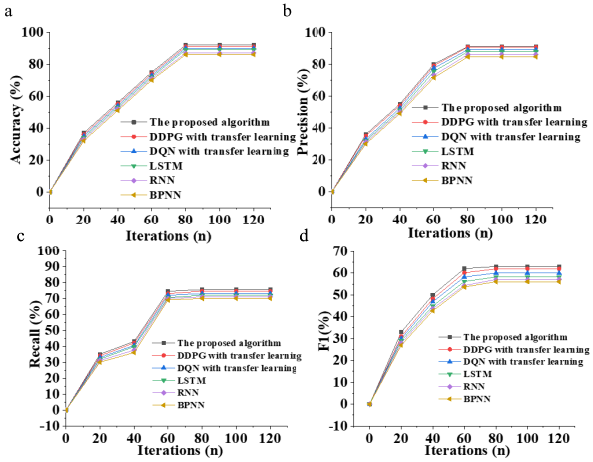

FIGURE 6. Influences on robot trajectory planning accuracy as the number of iterations increases under different algorithms. (Figures a, b, c, and d showed the comparison on accuracy, precision, recall, and F1 value, respectively.)

As shown in the figure above, the algorithm used in this study is compared with other neural network algorithms from the perspectives of accuracy, precision, recall, and F1 value. It can be found that the recognition accuracy of proposed algorithm reaches 92.53%, which is at least 5.47% higher than the model algorithm proposed by other scholars. In addition, the precision, recall, and F1 of the model algorithm in this study are 91.25%, 75.5%, and 63.51%, respectively. Compared with other algorithms, it is obvious that the precision, recall, and F1 value of the model algorithm in this study are at least 5.5% higher. Thus, compared with the advanced model algorithms proposed by other scholars in related fields, the robot trajectory planning based on the improved BPNN combined with Q-Learning constructed in this study is better.

B. EFFECTS OF VARIOUS ALGORITHMS

BPNN + Q-Learning algorithm is compared with DDPG combined with ML algorithm and DQN combined with ML algorithm to highlight the superiority of the proposed algorithm, and the results were shown in Figure 7 below:

Among the five algorithms, the BPNN + Q-Learning shows the shortest average calculation time per round (16365s); and the calculation time of BPNN, Q-Learning, DDPG combined ML algorithm, and DQN combined with ML algorithm are 19906s, 20078s, 18784s, and 18997s, respectively (as shown in Figure 6 above). DDPG combined ML algorithm and DQN combined with ML algorithm take less calculation time than BPNN and Q-Learning, but they require more time than BPNN + Q-Learning. The loss function of BPNN + Q-Learning has been stable after the sixth

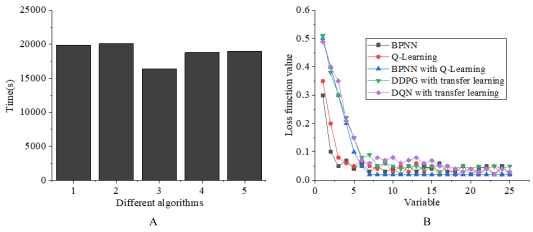

FIGURE 7. Comparison on performances of different algorithms. (Figure A shows the comparison results of running time. 1: BPNN; 2: Q-Learning; 3: BPNN +Q-Learning; 4: DDPG combined with ML algorithm; 5: DQN combined with ML algorithm; Figure B illustrates the change trend of loss function.)

variable, and the loss is significantly smaller than the other four functions.

This means that the selection and optimization of the algorithm is a key step in the robot path planning, which will determine the running time of the algorithm and whether the trajectory length of the robot path planning is the best.

C. EXPERIMENTAL RESULTS OF MOBILE ROBOT UNDER COMMON OBSTACLE ENVIRONMENT

The mobile robot executes corresponding movements according to the maximum Q values of the four outputs of BPNN, and performs movement trajectory planning in a discrete obstacle environment (as shown in Figure 8 below).

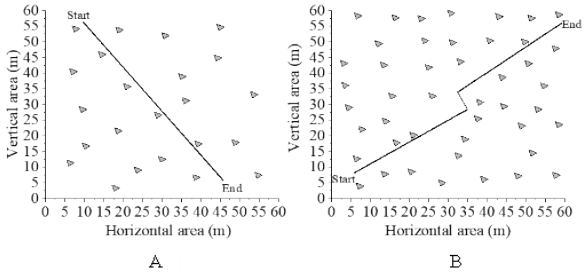

FIGURE 8. The trajectory planning of mobile robot in discrete obstacle environment.

There are fewer discrete obstacles in map A, and the gap is larger. The mobile robot can find the shortest movement trajectory, which is the best. At this time, the number of steps of the mobile robot after completing the planned trajectory is 100. The discrete obstacles in map B are significantly increased, and the gap interval is reduced. The mobile robot can still find the best movement trajectory and make the movement trajectory the shortest. At this time, the number of steps used by the mobile robot after completing the planned trajectory is 112. It suggests that under the robot path planning model of this study, the optimal path planning of the mobile robot will not be affected by the density of discrete obstacles and the size of the gap interval, realizing stable performance. It reveals that the algorithm used in this study can greatly promote the trajectory planning efficiency, enabling the robot to accurately find the best movement trajectory in

an environment with a small number of discrete obstacles or a large number of discrete obstacles. Then, the movement trajectory planning is tested in the continuous obstacle environment (as shown in Figure 9).

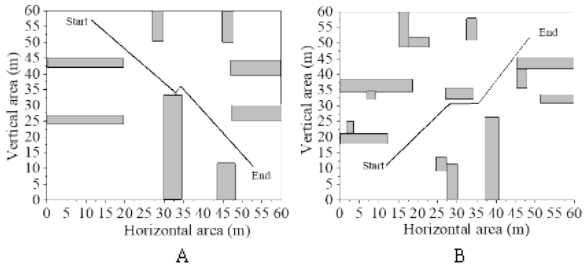

FIGURE 9. The trajectory planning of mobile robot in continuous obstacle environment.

The continuous obstacles in the above maps A and B constitute different environments similar to indoors. The mobile robot can perform the best trajectory planning in the two different environments. The number of steps of the mobile robot in the map A and map B is 111 and 123, respectively. It suggests that the mobile robot can perform optimal movement trajectory planning between any starting point and ending point, and learn a good trajectory without any collision.

D. EXPERIMENTAL RESULTS OF MOBILE ROBOT UNDER U-SHAPED OBSTACLE ENVIRONMENT

The mobile robot is placed in the U-shaped obstacle environment for the experiment of movement trajectory planning, and the results are shown in Figure 10.

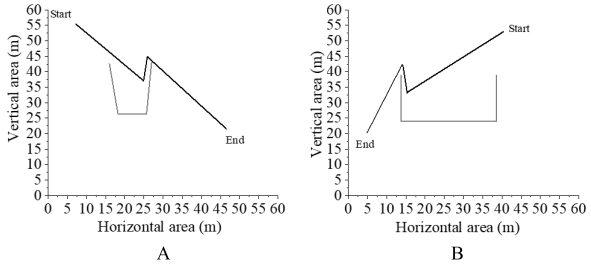

FIGURE 10. The trajectory planning of mobile robot in U-shaped obstacle environment.

Figures 10A and 10B indicate that the mobile robot can complete obstacle avoidance and path planning tasks in U-shaped obstacle environments of different sizes by using BPNN + Q-Learning algorithm. The numbers of steps in two maps are 132 and 125, respectively. BPNN + Q-Learning algorithm is not only in the optimal state of learning, but also includes generalized self-learning capabilities, which can generalize states that it does not include. Therefore, the mobile robot based on BPNN + Q-Learning algorithm can smoothly avoid U-shaped obstacle avoidance, and can plan the shortest path from the starting point to the ending point without collision, which can meet the requirements, and the effect is very satisfactory.

E. COMPARISON OF EXPERIMENTAL RESULTS OF VARIOUS ALGORITHMS IN DIFFERENT OBSTACLE ENVIRONMENTS

In the discrete obstacle, continuous obstacle, and U-shaped obstacle environments, the experimental results of BPNN, Q-Learning, and BPNN + Q-Learning are compared. The results are shown in Figures 11A,11B, and 11C.

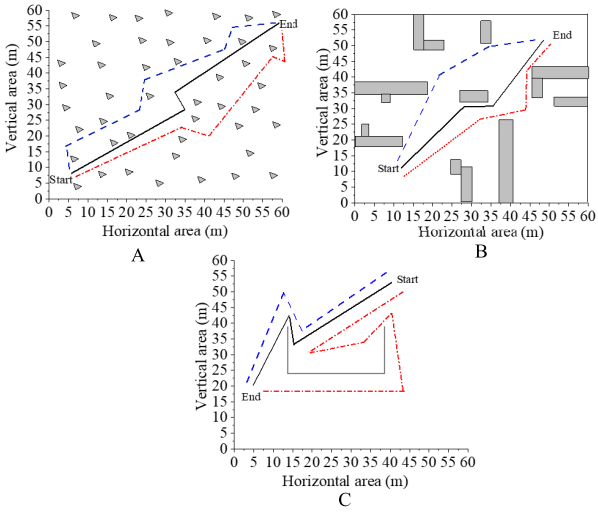

FIGURE 11. Experimental results of three algorithms under different obstacle environments.

(Note: Figure 10A: discrete obstacle environment; Figure 10B: continuous obstacle environment; and Figure 10C: U-shaped obstacle environment. Red dotted line shows the movement trajectory under BPNN algorithm; blue dotted line shows the movement trajectory under Q-Learning algorithm; and the solid line marks the movement trajectory under BPNN + Q -Learning algorithm.)

Figure 10A reveals that the lengths of the movement trajectory of mobile robot is 132, 130, and 112 under the three algorithms (BPNN, Q-Learning, and BPNN + Q-Learning) in a discrete obstacle environment, respectively. Figure 10B discloses that the lengths of movement trajectory of the mobile robot under three algorithms are 155, 160, and 123, respectively, under continuous obstacle environment. In the U-shaped obstacle environment, the lengths of the movement trajectory of the mobile robot are 167, 135, and 125 under the BPNN, Q-Learning, and BPNN + Q-Learning, respectively. Such results suggest that the movement trajectory of mobile robot under BPNN + Q-Learning algorithm at any obstacle environment is smaller than that under the BPNN and Q-Learning algorithm.

V. CONCLUSION

This study innovatively proposes a prediction model of BPNN + Q-Learning by combining the BPNN and Q-Learning algorithm to analyze the local movement trajectory of the robot. The results show that in different obstacle environments (discrete obstacles, continuous obstacles, or U-shaped obstacles), the mobile robot can plan the best

moving trajectory. It indicates that the mobile robots not only show better performance in dynamic and complex environments, but also can use the shortest number of steps to find the best planned trajectory. It suggests that the proposed algorithm in this study can be applied in robotics. However, there are some shortcomings in this study. The number of training samples is limited, and the samples contain some non-optimal state actions, which have an impact on the training effect of the BPNN. Therefore, the algorithm will be debugged in the future research to obtain more optimal training data, improve the training effect of the BPNN, and enable the mobile robot to find a more complete local estimation planning strategy.

COMPLIANCE WITH ETHICAL STANDARDS

Conflict of Interest: All Authors declare that they have no conflict of interest.

Ethical approval: This article does not contain any studies with human participants or animals performed by any of the authors.

Informed consent was obtained from all individual participants included in the study.

AUTHOR CONTRIBUTIONS

All authors listed have made a substantial, direct and intellectual contribution to the work, and approved it for publication.

REFERENCES

[1] L.-M. Semke and V. Tiberius, "Corporate foresight and dynamic capabilities: An exploratory study," *Forecasting*, vol. 2, no. 2, pp. 180–193, Jun. 2020.

[2] C. Diagne, J. A. Catford, F. Essl, M. A. Nuñez, and F. Courchamp, "What are the economic costs of biological invasions? A complex topic requiring international and interdisciplinary expertise," *NeoBiota*, vol. 63, p. 25, Mar. 2020.

[3] F. Porpiglia, E. Checcucci, D. Amparore, M. Manfredi, F. Massa, P. Piazzolla, D. Manfrin, A. Piana, D. Tota, E. Bollito, and C. Fiori, "Three-dimensional elastic augmented-reality robot-assisted radical prostatectomy using hyperaccuracy three-dimensional reconstruction technology: A step further in the identification of capsular involvement," *Eur. Urol.*, vol. 76, no. 4, pp. 505–514, Oct. 2019.

[4] J. Pan, X. Mai, C. Wang, Z. Min, J. Wang, H. Cheng, T. Li, E. Lyu, L. Liu, and M. Q.-H. Meng, "A searching space constrained partial to full registration approach with applications in airport trolley deployment robot," *IEEE Sensors J.*, vol. 21, no. 10, pp. 11946–11960, Mar. 2021.

[5] L. M. Gladence, C. K. Vakula, M. P. Selvan, and T. Y. S. Samhita, "A research on application of human-robot interaction using artifical intelligence," *Int. J. Innov. Technol. Explor. Eng.*, vol. 8, no. 9S2, pp. 2278–3075, 2019.

[6] L. Mary Gladence, M. Karthi, and T. Ravi, "A novel technique for multi-class ordinal regression-APDC," *Indian J. Sci. Technol.*, vol. 9, no. 10, pp. 1–5, Mar. 2016.

[7] F. H. Ajeil, I. K. Ibraheem, A. T. Azar, and A. J. Humaidi, "Grid-based mobile robot path planning using aging-based ant colony optimization algorithm in static and dynamic environments," *Sensors*, vol. 20, no. 7, p. 1880, Mar. 2020.

[8] A. J. Humaidi, I. K. Ibraheem, A. T. Azar, and M. E. Sadiq, "A new adaptive synergetic control design for single link robot arm actuated by pneumatic muscles," *Entropy*, vol. 22, no. 7, p. 723, Jun. 2020.

[9] F. H. Ajeil, I. K. Ibraheem, M. A. Sahib, and A. J. Humaidi, "Multi-objective path planning of an autonomous mobile robot using hybrid PSO-MFB optimization algorithm," *Appl. Soft Comput.*, vol. 89, Apr. 2020, Art. no. 106076.

[10] S. R. Jamalullah and L. M. Gladence, "Implementing clustering methodology by obtaining centroids of sensor nodes for human brain functionality," in *Proc. 6th Int. Conf. Adv. Comput. Commun. Syst. (ICACCS)*, Mar. 2020, pp. 1107–1110.

[11] P. K. Mohanty, "An intelligent navigational strategy for mobile robots in uncertain environments using smart cuckoo search algorithm," *J. Ambient Intell. Humanized Comput.*, vol. 11, no. 12, pp. 6387–6402, Dec. 2020.

[12] Y. Marchukov and L. Montano, "Multi-robot coordination for connectivity recovery after unpredictable environment changes," *IFAC-PapersOnLine*, vol. 52, no. 8, pp. 446–451, 2019.

[13] I. Giorgi, A. Cangelosi, and G. L. Masala, "Learning actions from natural language instructions using an ON-world embodied cognitive architecture," *Frontiers Neurorobot.*, vol. 15, p. 48, May 2021.

[14] J. Zheng, L. Gao, H. Wang, J. Niu, J. Ren, H. Guo, X. Yang, and Y. Liu, "Smart edge caching-aided partial opportunistic interference alignment in HetNets," *Mobile Netw. Appl.*, vol. 25, pp. 1842–1850, Jun. 2020.

[15] M. T. Singh, A. Chakrabarty, B. Sarma, and S. Dutta, "An improved on-policy reinforcement learning algorithm," in *Soft Computing Techniques and Applications*. Singapore: Springer, 2021, pp. 321–330.

[16] Z. Al-Qurashi and B. D. Ziebart, "Recurrent neural networks for hierarchically mapping human-robot poses," in *Proc. 4th IEEE Int. Conf. Robotic Comput. (IRC)*, Nov. 2020, pp. 63–70.

[17] G. Peng, C. Yang, W. He, and C. L. P. Chen, "Force sensorless admittance control with neural learning for robots with actuator saturation," *IEEE Trans. Ind. Electron.*, vol. 67, no. 4, pp. 3138–3148, Apr. 2020.

[18] G. A. Prasetyo, A. F. I. Suparman, Z. Nasution, E. H. Binugroho, and A. Darmawan, "Development of the gait planning for stability movement on quadruped robot," in *Proc. Int. Electron. Symp. (IES)*, Sep. 2019, pp. 376–381.

[19] Z. Liu and Y. Wang, "Trajectory planning for ground service robot," in *Proc. Chin. Control Decis. Conf. (CCDC)*, Jun. 2019, pp. 1511–1515.

[20] S. H. Lim and A. Autef, "Kernel-based reinforcement learning in robust Markov decision processes," in *Proc. Int. Conf. Mach. Learn.*, 2019, pp. 3973–3981.

[21] G. Marcjasz, B. Uniejewski, and R. Weron, "On the importance of the long-term seasonal component in day-ahead electricity price forecasting with NARX neural networks," *Int. J. Forecasting*, vol. 35, no. 4, pp. 1520–1532, Oct. 2019.

[22] W. Pan, L. Zhang, and C. Shen, "Data-driven time series prediction based on multiplicative neuron model artificial neuron network," *Appl. Soft Comput.*, vol. 104, Jun. 2021, Art. no. 107179.

[23] D.-W. Park, S.-H. Park, and S.-K. Hwang, "Serial measurement of S100B and NSE in pediatric traumatic brain injury," *Child's Nervous Syst.*, vol. 35, no. 2, pp. 343–348, Feb. 2019.

[24] F. Zhou, G. Lu, M. Wen, Y.-C. Liang, Z. Chu, and Y. Wang, "Dynamic spectrum management via machine learning: State of the art, taxonomy, challenges, and open research issues," *IEEE Netw.*, vol. 33, no. 4, pp. 54–62, Jul. 2019.

[25] Y. Jin, J. Guo, H. Ye, J. Zhao, W. Huang, and B. Cui, "Extraction of arecanut planting distribution based on the feature space optimization of PlanetScope imagery," *Agriculture*, vol. 11, no. 4, p. 371, Apr. 2021.

[26] H. Xie and Z. Wang, "Study of cutting forces using FE, ANOVA, and BPNN in elliptical vibration cutting of titanium alloy Ti-6Al-4V," *Int. J. Adv. Manuf. Technol.*, vol. 105, no. 1, pp. 5105–5120, 2019.

[27] A. A. Khater, A. M. El-Nagar, M. El-Bardini, and N. M. El-Rabaie, "Online learning based on adaptive learning rate for a class of recurrent fuzzy neural network," *Neural Comput. Appl.*, vol. 32, no. 12, pp. 8691–8710, Jun. 2020.

[28] M. A. Mojid, A. B. M. Z. Hossain, and M. A. Ashraf, "Artificial neural network model to predict transport parameters of reactive solutes from basic soil properties," *Environ. Pollut.*, vol. 255, Dec. 2019, Art. no. 113355.

[29] G. Karalekas, S. Vologiannidis, and J. Kalomiros, "EUROPA: A case study for teaching sensors, data acquisition and robotics via a ROS-based educational robot," *Sensors*, vol. 20, no. 9, p. 2469, Apr. 2020.

[30] S. Krul, C. Pantos, M. Frangulea, and J. Valente, "Visual SLAM for indoor livestock and farming using a small drone with a monocular camera: A feasibility study," *Drones*, vol. 5, no. 2, p. 41, May 2021.