International Journal of Mechatronics and Applied Mechanics
2026, Issue 25, Vol. II

INTELLIGENT TWIN MONITORING PLATFORM FOR WHEELED EQUIPMENT BASED ON GA-BP

*Fangli Qing, Pengzhan Zhao**
School of Intelligent Manufacturing, Chongqing City Vocational College, Chongqing, 402160, China

Abstract – Wheeled equipment is crucial for mobile support, but existing technologies suffer from low simulation accuracy and realism between physical equipment and virtual models. Furthermore, existing monitoring platforms lack sufficient interaction capabilities and have low linkage efficiency with physical robots, hindering efficient feedback and closed-loop control between virtual and real states. To address this, this study builds an intelligent twin monitoring platform using a wheeled mobile robot as an example and introduces a genetic algorithm to optimize the backpropagation neural network (GA-BP) for fault diagnosis of the robot's drive system. Findings demonstrate that the proposed GA-BP fault diagnosis algorithm achieves an accuracy of 94.7% on the test set, a 9.5% improvement over the traditional backpropagation neural network, and exhibits more stable identification capabilities across different fault types. Even under strong noise interference with a signal-to-noise ratio of 5dB, the algorithm maintains an accuracy of 77.8%, demonstrating good robustness. Simultaneously, the intelligent twin monitoring platform performs excellently in virtual-real synchronization, with the mean absolute position error controlled within 2cm during the robot's stable motion phase. In bidirectional control tests, the platform demonstrated a system latency of less than 235ms, verifying the platform's feasibility in real-time interaction and closed-loop control. In summary, the intelligent digital twin monitoring platform and the integrated GA-BP framework that were developed not only improve diagnostic accuracy but also enable bidirectional real-time interactive control, thereby validating the feasibility and effectiveness of digital twin technology in the intelligent operation and maintenance of wheeled equipment. This research provides an efficient intelligent technology for equipment fault diagnosis, promoting the digital development of equipment monitoring.

Keywords: Wheeled mobile robot, Digital twin, monitoring, Backpropagation neural network, Fault.

1. Introduction

Wheeled equipment is widely used in transportation, engineering operations and other fields. Its operational reliability directly affects operational efficiency and safety. However, in complex environments, such equipment often faces problems such as insufficient human-machine interaction or unpredictable sudden failures, which affect its effectiveness. The core of digital twin (DT) technology lies in building a virtual model that is synchronized with the physical entity in real time to support simulation analysis and decision optimization. It provides an effective way to improve the reliability and maintainability of wheeled equipment [1-2]. Wheeled mobile robots (WMR) have shown great application value in manufacturing, medical care and logistics due to their high efficiency, flexibility and strong adaptability [3]. Therefore, research has been conducted to explore its monitoring application [4]. The key applications and achievements of DT technology in improving the perception, interaction, control and operation performance of robots are reflected in the following aspects. For example, Li Y designed an intelligent inspection robot system that integrates the Internet of Things and improved Simultaneous Localization and Mapping (SLAM) algorithm for urban traffic problems. The results showed that the system improved the computational efficiency to 80% and the modeling accuracy by 23.3% [5].To systematically improve the perception ability of collaborative robots, Müller et al. proposed a method for quantifying the state of situational

dx.doi.org/10.17683/ijomam/issue25/v2.6

awareness using DT technology. The results showed that the method is practical [6]. Ye et al. verified the high efficiency and flexibility of the technology by using DT and human-machine collaboration technology to address the problems of low assembly efficiency and poor flexibility of complex irregular building structures [7]. Zhao et al. constructed a DT system for vehicle-like mobile robots that can realize virtual-real tracking control. They realized bidirectional motion mapping through a nonlinear extended state observer and a backstepping and integral sliding mode controller. The results showed that the system was effective in virtual-real tracking control [8].

With the advancement of deep learning (DL) technology, numerous researchers have provided effective technical solutions for the status monitoring and high-precision fault diagnosis of key components of various industrial robots. For example, Bilal et al. introduced an Internet of Things architecture and a hybrid one-dimensional convolutional neural network (1D-CNN) method for online detection of joint faults in industrial robotic arms. The findings demonstrated that the fault detection accuracy of this method on the UR16e robotic arm reached 99.03% [9]. To improve the fault diagnosis accuracy of heavy industrial robots, Wang et al. built a dynamic simulation model with joint flexibility through the Euler-Lagrange method to generate fault samples, and used a long short-term memory network (LSTM) and CNN (LSTM-CNN) composite model for training. The results showed that the technology was effective [10]. Polenghi A et al. built a predictive health management framework based on a hybrid unsupervised artificial intelligence algorithm for monitoring abnormal robot trajectories. The results showed that the framework could effectively adapt to different trajectories and report abnormal equipment status in a timely manner [11]. To address the challenges of weak fault feature extraction and high-precision fault diagnosis of the main bearing of tunnel boring machine under strong noise and complex working conditions, Zhong et al.introduced a weighted domain adaptation network (ResNet-Attention) that integrates residual denoising and multiscale attention. The findings demonstrated that the classification accuracy of this method was improved by up to 43.13% [12].

In summary, existing research has yielded promising results. However, the application of existing DT technologies in WMR monitoring suffers from the following three main shortcomings. First, virtual models are primarily used as passive mapping tools and lack the ability for bidirectional, closed-loop interaction with physical entities. Second, fault diagnosis algorithms rely mainly on single neural networks or traditional machine learning methods, leaving room for improvement in terms of parameter adaptability and accuracy in identifying multiple fault types. Third, the level of integration between the monitoring platform and the diagnostic module is low, and a closed-loop operation and maintenance chain covering the "perception–diagnosis–feedback" stages has not yet been established. To address these shortcomings, this study has made the following methodological advancements: First, this study has developed a framework for an Intelligent Twin Monitoring Platform (ITMP) that supports bidirectional command interaction between virtual and physical entities. Concurrently, this research proposed the GA-BP algorithm, which uses genetic algorithms (GA) to optimize the initial parameters of a Backpropagation Neural Network (BPNN) to drive system fault diagnosis, and embed the diagnostic functionality into the twin service platform to achieve integrated operation of state synchronization, fault identification, and strategy feedback. Through this design, this study has achieved distinct methodological advancements compared to existing work across three dimensions: interaction architecture, optimization strategies, and system integration.

2. Research Design

2.1 Overall Design and Modeling of ITMP in WMR

- ***Platform Overall Architecture Design***

The study combined high-fidelity 3D modeling with a high physical realism simulation engine to create a virtual model that closely matches the real-world robot, thereby ensuring the accuracy and simulation realism of the virtual robot. Specifically, SolidWorks and 3D Max software were used for 3D modeling, while Unreal Engine 4 and AirSim were selected as the simulation engines [13]. Based on these settings, the study built the ITMP framework, as shown in Figure 1.

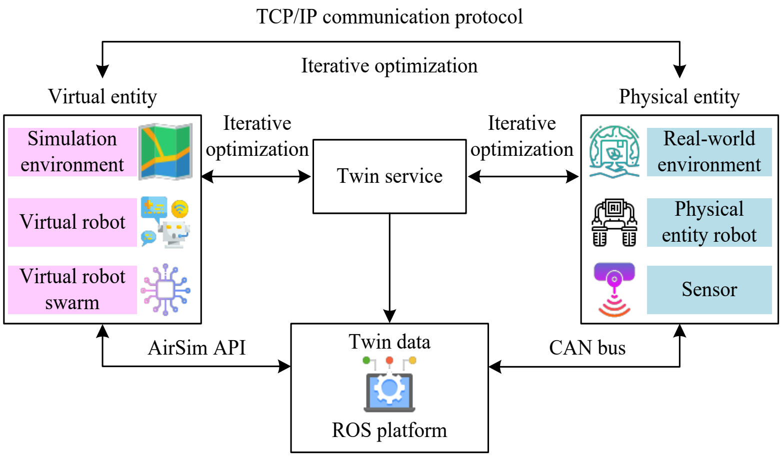


Figure 1: The ITMP framework

In Figure 1, the framework consists of virtual entities, twin services, physical entities and iterative optimization mechanisms. Virtual entities include real scenes simulated by AirSim, virtual robots and their clusters. Virtual entities are digital representations of physical entities, which can quickly display the real environment to a certain extent, making it convenient for testing and verification. Physical entities refer to the actual physical robots, sensors and real working environments that are actually displayed. They belong to the execution end of the system and are mainly responsible for completing tasks in reality and transmitting data to the virtual end. In order to solve the problems of insufficient interaction capabilities and low linkage efficiency, this study establishes a real-time, bidirectional data channel between virtual entities and physical entities through the Robot Operating System (ROS) framework and TCP/IP communication protocol. ROS is a flexible framework designed for robot development, which is free and open source. It provides functions such as hardware abstraction, driver management, message passing and package management [14]. Virtual entities and physical entities interact directly through the TCP/IP communication protocol to realize the state synchronization and instruction transmission of virtual and real robots. Simulation data from virtual entities is transmitted to ROS via the AirSim API, while sensor and control data from physical entities are transmitted to ROS via the Controller Area Network (CAN) bus. ROS then distributes the data to the twin service for processing. Based on the fused virtual and physical data, the twin service optimizes algorithms and control strategies, and pushes the optimization results back to the virtual and physical entities via ROS, forming an iterative optimization loop.

- ***Kinematic Model of WMR***

To ensure that the virtual robot in the simulation engine can move like a physical robot, the research requires providing precise motion control algorithms for the virtual robot. Each wheel of the selected WMR is typically driven by a motor, and a planetary gearbox or harmonic reducer is used to increase torque. These components also include gears and bearings, which are the areas of highest mechanical wear in the robot. The kinematic model of this type of robot is shown in Figure 2.

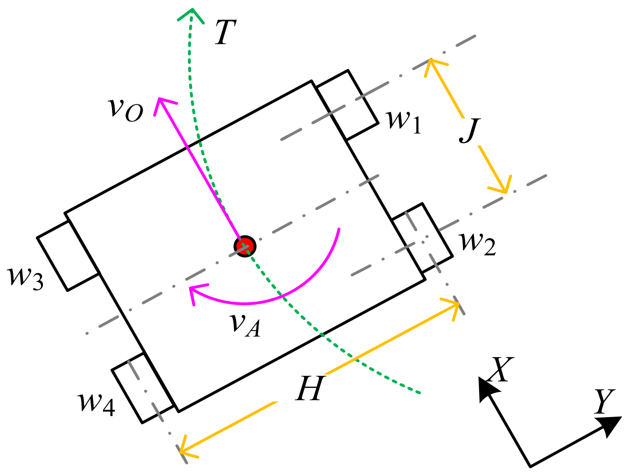


Figure 2: Kinematic model of the robot

In Figure 2, the chassis of the WMR is rectangular, T represents the robot's motion trajectory. H represents the distance between the robot's left and right wheels. J represents the distance between the robot's front and rear wheels. This robot uses a differential drive principle, meaning the left and right wheels can independently control their rotational speed. The relationship between the wheel speeds is shown in equation (1).

$$\begin{cases} w_R = w_1 = w_2 \\ w_L = w_3 = w_4 \end{cases} \tag{1}$$

In equation (1), w_L and w_R represent the rotational speeds of the left and right sides. w_1 and w_2 represent the rotational speeds of the right front wheel and the right rear wheel, respectively. w_3 and w_4 represent the rotational speeds of the left front wheel and the left rear wheel. Let the robot 's attitude angle vector in the local coordinate system $L(x, y, z)$ be Φ, as shown in equation (2).

$$\Phi = [\varepsilon, \alpha, \gamma]^T \tag{2}$$

In equation (2), ε is the rotation angle of the robot around its own X-axis. α is the rotation angle of the robot around its own Y-axis. γ is the azimuth angle. The study only considers the robot's planar motion, that is, it does not consider ε and α, only considers γ. Assuming that longitudinal and lateral sliding are not considered, the study uses the base point method to calculate the speed of the four wheels [15]. The linear velocities of the left and right wheels in the X direction are w_{LX} and w_{RX}, respectively, and their calculations are shown in equation (3).

$$\begin{cases} v_{RX} = v_{1X} = v_{2X} = v_{OX} + (H/2)v_A \\ v_{LX} = v_{3X} = v_{4X} = v_{OX} - (H/2)v_A \end{cases} \tag{3}$$

In equation (3), v_{OX} represents the velocity of the robot's center of mass in the X direction. v_{AX} represents the angular velocity of the robot in the X direction. In the coordinate system $L(x, y, z)$, the X coordinate of the instantaneous center is D, then equation (4) is given.

$$D = -v_{OY}/w \tag{4}$$

In equation (4), v_{OY} represents the velocity of the robot's center of mass in the Y direction. The velocity components of the four wheels in the Y direction are calculated as shown in equation (5).

$$\begin{cases} v_{1Y} = v_{3Y} = v_{OY} - Dv_A \\ v_{2Y} = v_{4Y} = v_{OY} + Dv_A \end{cases} \tag{5}$$

In equation (5), v_A represents the angular velocity of the robot. In the robot coordinate system, it is assumed that the tires do not slip in the forward and backward directions, and the rolling radius of the wheel is always equal to its own geometric radius. It is also assumed that the robot does not drift laterally during the movement, and its direction of movement is always consistent with the orientation of the vehicle body. Under these two assumptions, the kinematic model used to describe the relationship between the overall motion of the robot and the rotational speed of the left and right wheels is shown in equation (6) [16].

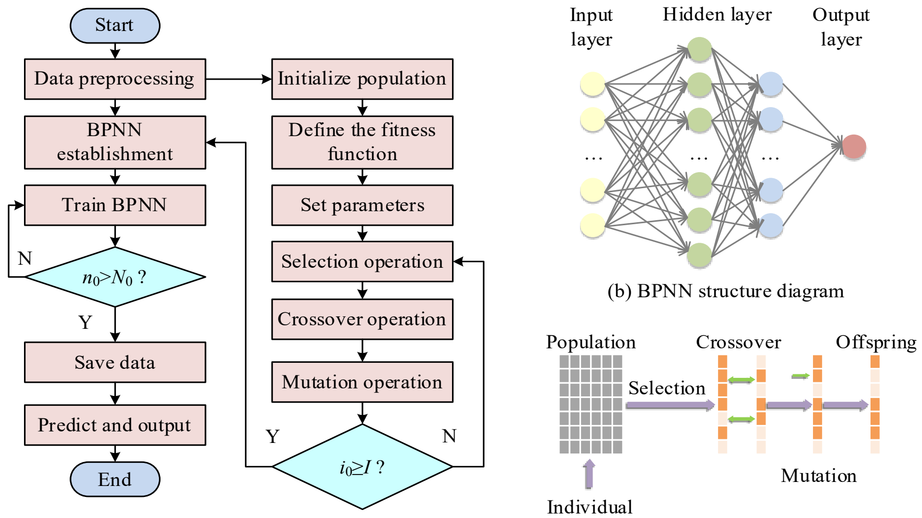
$$\begin{bmatrix} v_{OX} \\ v_{OY} \\ \gamma \end{bmatrix} = \begin{bmatrix} r/2 & r/2 \\ 0 & 0 \\ -r/H & r/H \end{bmatrix} \begin{bmatrix} w_L \\ w_R \end{bmatrix} \tag{6}$$

In equation (6), r is the wheel radius. Using the kinematic model as the core carrier of virtual-real interaction, on the one hand, sensor feedback data from the physical robot is collected to drive the virtual model to complete synchronous motion. On the other hand, control commands generated in the virtual space are precisely sent to the real robot to execute corresponding actions.

2.2 WMR Fault Detection based on GA-BP

After designing ITMP,to improve the efficiency of fault detection, the GA-BP algorithm module was embedded in the twin service. Although traditional BPNNs possess nonlinear fitting capabilities in fault diagnosis, their initial weights and thresholds are set randomly, which can easily cause the model to get stuck in a local optimum, and their convergence rate is significantly affected by the choice of initial values. Genetic algorithms, with their global search capabilities, are well-suited for searching for near-optimal initial parameter combinations in high-dimensional parameter spaces [17]. Therefore, this study designed a GA-BP algorithm aimed at improving the stability and accuracy of fault diagnosis. Specifically, this study uses GA to globally optimize the initial weights and thresholds of BPNN, and synchronously encodes the number of hidden layer nodes with network parameters in the encoding design, so that the optimization space covers the configuration level of the network structure. This design differs from the common practice in existing research that only uses GA as a parameter replacement tool, allowing the optimization process to have both structural search and parameter tuning functions. Compared to heuristic algorithms such as particle swarm optimization, GAs demonstrate better adaptability when handling mixed discrete and continuous parameter optimization problems and are less susceptible to interference from the initial population distribution; thus, this study prioritized the use of GAs as the optimization strategy.

The complete technical description of the GA-BP algorithm is shown in Figure 3.



(a) Flowchart of GA-BP algorithm (b) BPNN structure diagram (c) Schematic diagram of GA core process

Figure 3: Complete technical illustration of GA-BP algorithm

Figure 3(a) shows the workflow of the GA-BP algorithm. The study collects various data of WMR operation based on ROS, such as motor current, voltage and speed. The collected data will contain noise and outliers. These data are preprocessed by cleaning, normalization and feature extraction. Then, the optimal initial weights and thresholds are found for BPNN to avoid BPNN getting trapped in local optima. The study performs GA optimization operation. Specifically, this operation refers to initializing the population, encoding the weights and thresholds of BPNN into individuals of GA to form the initial population. Then, the fitness function is defined as the recognition accuracy of fault samples to evaluate the quality of each parameter combination. In order to transform the BPNN error minimization problem into the GA fitness maximization problem, the study constructs the fitness function of the following equation (7) [18].

$$F=\frac{1}{E+\delta}, E=\frac{1}{2N_1}\sum_{p=1}^{N_1}\sum_{k=1}^{M_1}\left(Y_{pk}-O_{pk}\right)^2 \qquad (7)$$

In equation (7), E is the calculated BPNN prediction error. δ is a very small positive number to prevent the denominator from being zero. N_1 is the total number of training samples. Y_{pk} is the true fault label of the p th sample. O_{pk} is the actual output of the network. Next, selection, crossover, and mutation operations are performed to iteratively select a better combination of parameters through simulated natural selection. When the iteration count i_0 reaches the set value I, the optimal combination of weights and thresholds is output. The optimal parameters obtained by GA are used to initialize the BPNN, and then training and prediction are performed. According to the needs of robot fault detection, the structure of the input layer (IL) (such as motor current, speed, etc.), hidden layer (HL), and output layer (OL) are determined. The BPNN is trained using the preprocessed robot running data and the optimal initial parameters given by GA. If the number of training times n_0 reaches the maximum number of training times N_0, training is stopped. The trained model is saved, and then the model can be used to detect faults in the real-time collected WMR data and output the fault type or warning. Figure 3(b) is a BPNN structure diagram. The network includes an IL, anHL, and an OL. The IL receives the robot's running features, and then the HL performs a nonlinear transformation on the input features to extract deep features related to the fault. Finally, the probability of the fault type is output in the OL. Assuming the number of nodes in the IL is n and the number of nodes in the HL is U, the calculation from the IL to the HL is shown in equation (8) [19].

$$y_j=f_I\left(\mu_j+\sum_{m=t-n}^{t-1}\mu_{jm}y_m\right), \quad \left(0\le\mu_j,\mu_{jm}\le1\right) \qquad (8)$$

In equation (8), y_m and y_j represent the inputs of the IL and the HL, respectively. f_I represents the activation function of the HL. μ_j is the threshold of the HL. μ_{jm} is the network weight of the OL. The calculation from the HL to the OL is shown in equation (9) [20].

$$y_t=f_O\left(\lambda_O+\sum_{j=1}^{U}\lambda_{Oj}y_j\right), \quad \left(0\le\lambda_O,\lambda_{Oj}\le1\right) \qquad (9)$$

In equation (9), y_t represents the predicted value of point t. f_O is the activation function of the OL. λ_O is the threshold of the OL. λ_{Oj} is the network weight of the HL. Figure 3(c) is a visualization of the core process of GA. The population consists of multiple individuals representing BPNN parameters. Excellent individuals are selected based on fitness, and then parts of the genes of two parent individuals are exchanged to generate offspring. Parts of the genes of the offspring are randomly changed to increase population diversity.

The newly generated individuals become the next generation population, and iterative optimization continues. The operating environment of WMR is highly variable, and the combined GA-BP model is more robust, maintaining high detection accuracy even under noise interference. The driving system of WMR exhibits nonlinear and strongly coupled fault characteristics. To train and test the GA-BP fault diagnosis model, six common driving system faults were simulated in the AirSim simulation environment, as shown in Table 1.

Table 1: Target fault types of the drive system of wheeled robots

Fault number	Fault name	Subsystem to which it belongs
F0	Normal state	/
F1	Early wear of motor bearing	Motor-mechanical
F2	Partial tooth breakage of the reducer gear	Transmission-Mechanical
F3	Motor driver single-phase open circuit	Power electronics
F4	One-sided tire pressure loss	End execution
F5	Intermittent loss of encoder signal	Control-Sensor

3. Results and Analysis

3.1 Experimental Setup

In this study, operational data were collected for each of the six fault types listed in Table 1 within the AirSim simulation environment, with 2,000 data sets collected for each fault type, for a total of 12,000 data sets. The data collection process covered a range of rotational speeds and load conditions to enhance the diversity of the samples. The samples were then randomly divided into training, validation, and test sets in a 7:1.5:1.5 ratio. The validation set was used for early stopping to prevent overfitting, while the test set remained excluded throughout the training and parameter selection processes. Data preprocessing included imputing missing values with the mean, removing outliers based on the 3σ principle, and standardizing the data using Z-scores to eliminate the effects of different units of measurement. To reduce bias introduced by random partitioning, all experiments were repeated 10 times, and the average performance and standard deviation were reported. The physical robot used in this study was the Clearpath Jackal, and its main technical parameters are shown in Table 2.

Table 2: Main technical parameters of the robot

Parameter	Specification
External dimensions (length × width × height)	508 mm × 430 mm × 250 mm
Vehicle weight	17 kg
Maximum load	20 kg (flat terrain, limit 50 kg) / 10 kg (rough terrain)
Battery type	270Wh Lithium Battery
Motor	4 × 120W DC brushless motors
Gearbox	16:1 Planetary gearbox
Steering type	4-wheel differential steering
Maximum speed	2.0 m/s (6.6 ft/s)
Minimum turning radius	0 m (in-place turning)
Maximum climbing ability	30° (flat terrain) / 15° (rough terrain)
Minimum ground clearance	127 mm
Protection class	IP65 (Waterproof and dustproof)
Control mode	Open-loop, speed & motion commands, ROS control, remote control, autonomous navigation
Communication interface	Ethernet, Wi-Fi, USB 3.0, CAN
Navigation method	Supports LiDAR SLAM, Visual SLAM, GPS/RTK positioning (cm-level accuracy)
Operating temperature	-19 °C ~ +45 °C
Software & API	ROS (Noetic/Melodic), Mathworks, Gazebo 3D Simulation Environment

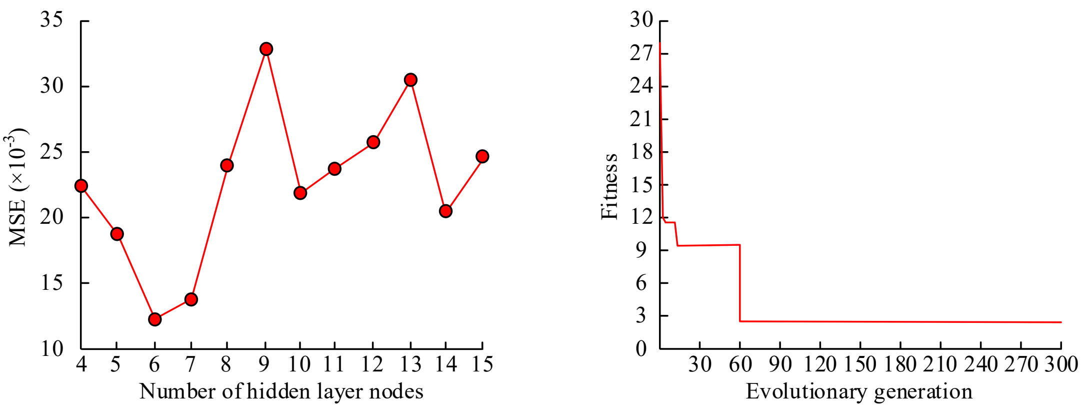
The ITMP software environment was configured as follows: Operating system: Ubuntu 18.04; Robot middleware: ROS Melodic; Virtual simulation environment: Unreal Engine 4.27.2 and AirSim v1.7.0; Algorithm development and some data processing were performed in Visual Studio 2019.

3.2 Performance of GA-BP Fault Diagnosis Algorithm

- ***GA-BP Parameter Settings and Fitness Results***

The specific implementation parameters of the GA-BP algorithm were set as follows: BPNN adopted a single HL structure with 6 OL nodes. The number of HL nodes was determined by grid search optimization. The activation function was Sigmoid, the learning rate was set to 0.05, and gradient descent was used for optimization. The population size of the GA part was set to 50, the max number of generations was 300, and the crossover probability and mutation probability were 0.9 and 0.1. To verify the effectiveness of setting the number of HL nodes in BPNN, the mean squared error (MSE) results of the test set corresponding to different numbers of nodes were studied and analyzed.

In addition, the change of population fitness during GA iteration was also analyzed. The outcomes of the two experiments are presented in Figure 4. In Figure 4(a), when the number of nodes was 6, the MSE reached the lowest value (1.21×10-2). At this time, the fault detection accuracy of the model was the highest. When the number of nodes was 4, the MSE was 2.23×10-2, and when the number of nodes was 9, the MSE was the highest (3.28×10-2). When the number of nodes was 10-15, the MSE exceeded 2.00×10-2. Too few or too many nodes resulted in a high MSE, indicating underfitting or overfitting. Therefore, the optimal network structure for GA-BP was 6 HL nodes. In Figure 4(b), the fitness rapidly decreased to 12 in the initial 0-10 generations. From generations 10-15, the fitness decreased slightly to 9. From generations 15-60, the fitness stabilized at 9. After generation 60, the fitness stabilized at 2.8, entering a plateau and no longer changing. This indicated that GA found a better combination of BPNN parameters early on, and subsequent iterations did not further improve it, satisfying the algorithm's termination condition. The plateau also verifies the efficiency of GA in global optimization, avoiding ineffective iterations.



(c) MSE comparison

(c) Fitness curve

Figure 4: Changes in MSE and fitness curve with varying numbers of HL nodes

- ***Comparison and Analysis of GA-BP algorithms***

To confirm the effectiveness of the GA-BP algorithm, it was compared with conventional BPNN, Particle Swarm Optimization (PSO) BP (PSO-BP), Support Vector Machine (SVM), and Random Forest (RF). To ensure the fairness of the comparative experiments, all models were trained and evaluated on the same training and test sets, and their key hyperparameters were optimized using a grid search strategy. In PSO-BP, the particle swarm size was set to 50, and the inertia weight was adjusted using a linearly decreasing method. For the SVM, the radial basis kernel was selected, and both the regularization coefficient and the kernel parameter were searched within the range [0.1, 100]. For the RF, the number of decision trees was searched within the range [50, 500], with a maximum depth limited to 10. All models were run in the same software and hardware environment, and the results were averaged after each model was run independently five times to reduce the impact of randomness on the experimental results. The performance comparison of different fault diagnosis models on the test set is shown in Table 3. In Table 3, GA-BP outperformed the comparison models in all four metrics. Its accuracy (94.7%) was 9.5% higher than BPNN, demonstrating the effectiveness of GA in optimizing BPNN by avoiding local optima. Compared to PSO-BP, GA-BP's macro precision (93.8%) improved by 3.3%. This was because GA's global search strategy on the test set was more effective than PSO in finding the optimal initial point for BPNN. GA-BP's macro recall and macro F1 score were 93.0% and 93.4%, respectively. SVM and RF's macro F1 scores were 86.2% and 88.7%, respectively. In contrast, GA-BP had a stronger automatic feature learning capability, which could extract more effective deep nonlinear representations for fault diagnosis from the original features, thereby reducing the reliance on manually designed features.

Table 3: Performance comparison of various fault diagnosis models on the test set

Model	Accuracy (%)	Macro precision (%)	Macro recall (%)	Macro F1 score (%)
BPNN	85.2	83.5	82.1	82.8
GA-BP	94.7	93.8	93.0	93.4
PSO-BP	92.1	90.5	90.0	90.2
SVM	88.9	87.0	85.4	86.2
RF	90.3	89.1	88.3	88.7

To analyze the accuracy of the proposed GA-BP model in identifying fault types, it was compared with the traditional BPNN. The confusion matrix outcome are presented in Figure 5. Figure 5(a) indicates the confusion matrix of BPNN. Its accuracy in identifying these six fault types ranged from 77% to 89.0%, with the highest accuracy for F1 (89.0%). Its accuracy for F2 was the lowest, at only 77.0%, with 12.0% of F2 samples being misclassified as F4. Figure 5(b) shows the confusion matrix of GA-BP. The overall accuracy of GA-BP ranged from 89.0% to 98.0%, and its identification rate for different faults was significantly higher than that of BPNN. Its accuracy for F1 reached the highest (98.0%). The accuracy for F2 improved from 77.0% to 89.0%, and the probability of misclassifying as F4 decreased from 12.0% to 4.0%. Therefore, GA-BP could effectively improve the accuracy and robustness of fault detection in WMR drive systems.

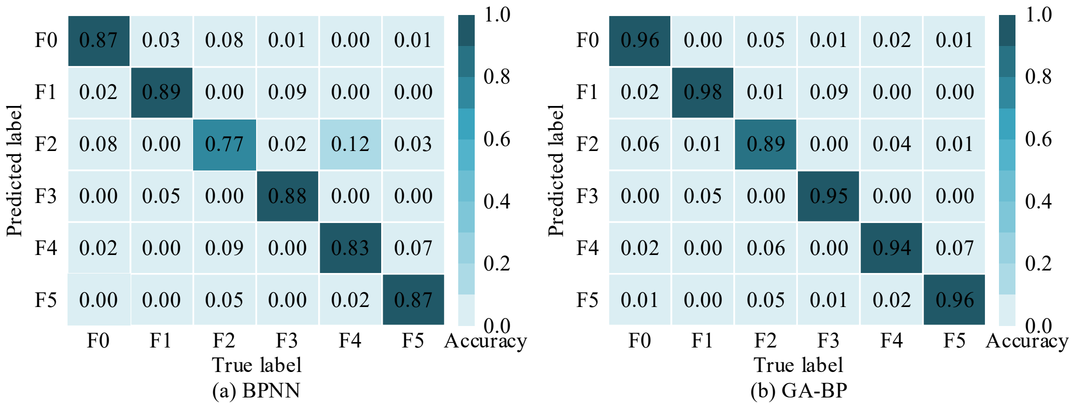


Figure 5: Confusion matrix of BPNN and GA-BP for different fault types

- ***Robustness Testing***

In real-world robotic operating environments, sensor signals are often affected by factors such as electromagnetic interference, mechanical vibrations, and communication packet loss, resulting in the superposition of noise of varying intensities. To simulate such non-ideal operating conditions, this study added Gaussian white noise with varying signal-to-noise ratios (SNR) to the sensor signals in the test set to evaluate the noise immunity of each fault diagnosis model under noisy conditions. A value of 5 dB corresponds to a high-noise scenario (where the signal power is approximately three times the noise power) and is used to test the robustness limits of the models under extreme noise conditions. Among them, the study introduced the 1D-CNN model from reference [9] and the ResNet-Attention model from reference [12] for comparison. The comprehensive performance comparison of different models is presented in Figure 6. Figure 6(a) shows the accuracy changes of different models under different noise levels. Under 5dB noise, the accuracy of GA-BP (77.8%) was 22.5% higher than that of BPNN (55.3%) and 13.9% higher than that of SVM (63.9%). Compared with GA-BP, 1D-CNN and ResNet-Attention had higher accuracy. Both maintained a high level of accuracy under all noise interference. Under 5dB noise, the accuracy of GA-BP was 7.5% lower than that of ResNet-Attention (85.3%). Figure 6(b) compares the training time and single-sample inference time of the models. The training times for 1D-CNN and ResNet-Attention were 120 min and 180 min, respectively. Their single-sample inference times were 2.5 ms and 5.0 ms. SVM had the fastest training speed (5 min) and the smallest inference latency (0.5 ms). GA-BP achieved a good balance between training cost (25 min) and inference speed (1.2 ms). Figure 6(c) compares the number of parameters and test accuracy of the models. ResNet-Attention had 5.0×104 parameters,and its accuracy advantage was not significant compared to 1D-CNN (1.5×104) and GA-BP (8.5×104), while its complexity and training time increased exponentially.

With the same parameter scale of 8.5×104, the accuracy of GA-BP (94.7%) was significantly higher than that of BPNN (85.2%). Compared to PSO-BP, GA-BP achieved higher accuracy with shorter training time, demonstrating the optimization efficiency of GA in fault diagnosis tasks.

In comparison, GA-BP had only 17% of the parameters of ResNet-Attention, resulting in faster training and inference speeds. Overall, the proposed GA-BP exhibited good comprehensive performance, achieving a good balance between the high performance of DL and the lightweight nature of traditional methods.

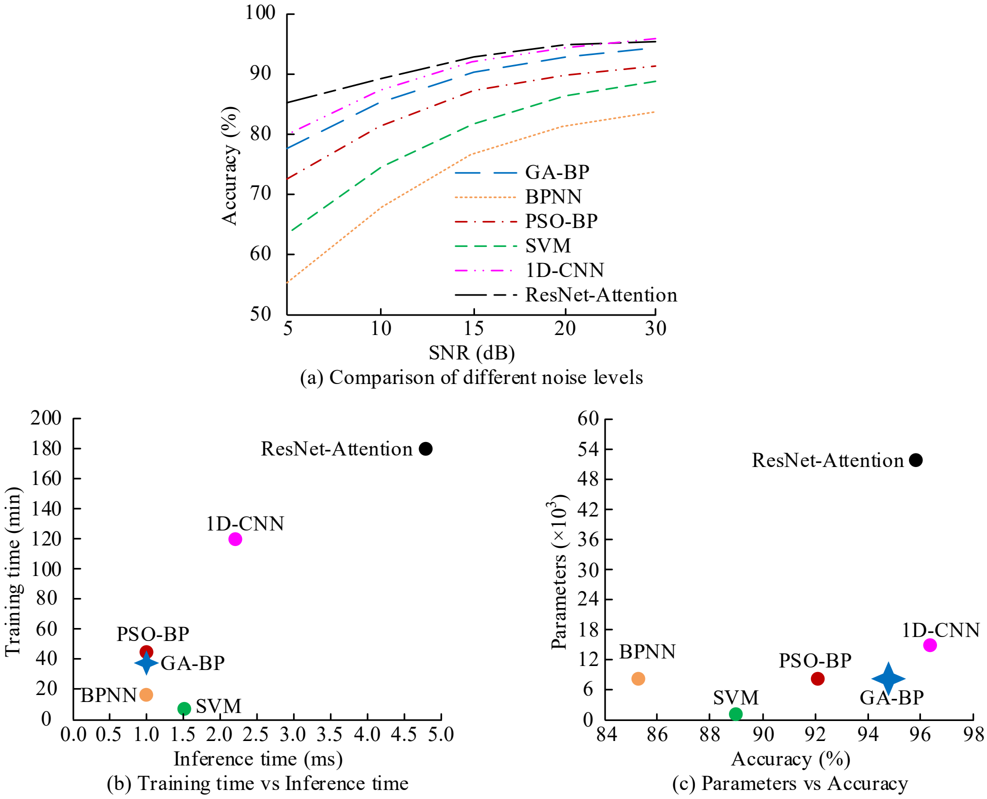


Figure 6: Comparison of comprehensive performance of different models

- ***Multi-Condition Test Analysis***

To further evaluate the generalization capability of the proposed GA-BP algorithm under different operating conditions, this study additionally set up three types of test scenarios in the simulation environment: variable-speed motion (0.2–1.0 m/s sinusoidal speed variation) , heavy-load simulation (load increased to 15 kg), and road gradient (5° and 10° slopes). Operational data were collected for each failure type and input into the trained GA-BP model for testing. The test results are shown in Table 4.

As shown in Table 4, the model's accuracy was approximately 91.2% under variable-speed conditions, approximately 89.7% under heavy-load conditions, and ranged from 86.8% to 88.3% under slope conditions, all of which remained at a relatively high level.

This indicates that the model possesses a certain degree of adaptability to changes in operating conditions. Among these, the performance decline under slope conditions was the most pronounced, primarily because the mapping relationship between motor current and wheel speed shifted during slope driving, resulting in inconsistencies between the distribution of certain features and the training set.

Table 4: Fault diagnosis accuracy of GA-BP algorithm under different operating conditions (unit:%)

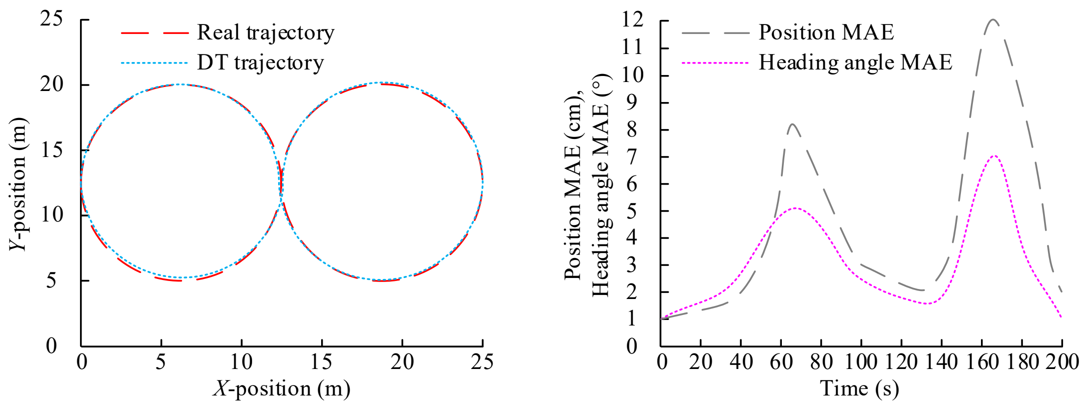
Operating conditions	F0	F1	F2	F3	F4	F5	Average accuracy rate
Constant speed on level ground (0.5 m/s)	96.0	98.0	89.0	94.0	93.0	97.0	94.5
Variable speed (0.2–1.0 m/s)	93.5	95.0	85.5	91.0	90.0	92.0	91.2
Heavy load (+15 kg)	92.0	93.5	83.0	89.5	88.0	92.0	89.7
5° incline	91.0	92.0	82.0	88.0	86.5	90.0	88.3
10° incline	90.0	91	80.5	86.5	84.0	88.5	86.8

3.3 ITMP Functional Verification

- ***Verification of Virtual-Real Synchronization Accuracy***

To verify the synchronization between the real and virtual states of the ITMP, the robot was controlled by a handheld remote controller to execute a preset "∞" shaped trajectory at a speed of approximately 0.5 m/s. The error between the robot's real trajectory and the ITMP's DT trajectory was recorded. The comparison of different trajectories and the mean absolute error (MAE) time-series curves are shown in Figure 7. Figure 7(a) shows the synchronization comparison of the two trajectories. The DT trajectory and the real trajectory showed a high degree of consistency along the overall "∞" shaped path, which verified the reliability of the ITMP's state synchronization function.

At high-speed turns and path intersections, obvious trajectory separation was observed. This was because the theoretical kinematic model did not consider the longitudinal and lateral slippage of the tires, leading to model inaccuracy under high center of acceleration conditions. Figure 7 (b) shows the position MAE and heading angle MAE curves over time. During the 200s test period, the maximum instantaneous position error was 12.0 cm, and the maximum instantaneous heading angle error was 6.7°. During the uniform straight-line segment from 0 to 20 seconds, both errors remained at low levels, with position MAE < 2.0 cm and heading angle MAE < 2°, indicating that the platform had good synchronization accuracy under stable conditions.

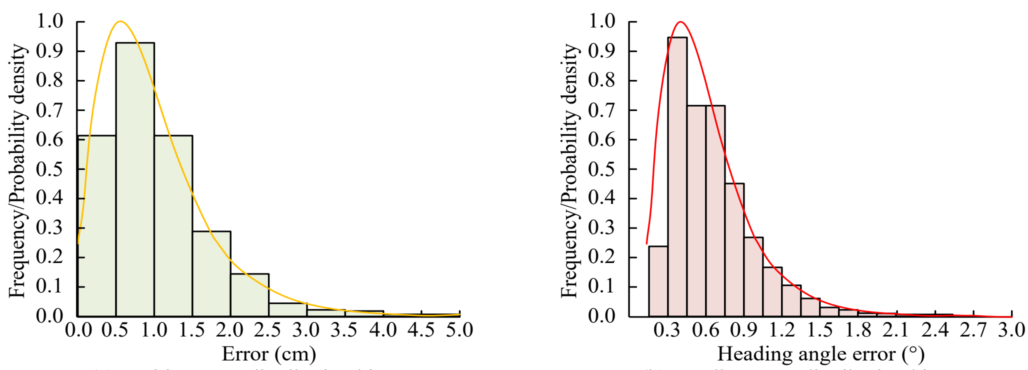


(a) Trajectory synchronization comparison (c) Change in error over time

Figure 7: Comparison of different trajectories and MAE time series curve

To analyze the error distribution characteristics in depth, typical uniform linear motion and uniform turning motion segments were extracted from the "∞" shaped trajectory, and their position and heading angle errors were calculated respectively. The distribution histograms of the two errors are shown in Figure 8. In Figure 8(a), the position error of the robot in the linear motion segment was mainly concentrated in the range of 0~2cm, with the peak appearing at 0.6cm and a probability density exceeding 0.9.

The position error distribution histogram was right-skewed, with a maximum error of 5.0cm. In Figure 8 (b), the heading angle error of the robot in the turning motion segment was highly concentrated in the range of 0~1.5 °, with the peak appearing at 0.45 ° and a probability density close to 1.0. Therefore, during the steady motion phase, ITMP exhibited centimeter-level position synchronization accuracy and sub-degree-level heading synchronization accuracy.



(a) Position error distribution histogram (b) Heading error distribution histogram

Figure 8: Distribution histogram of position error and heading angle error

- ***Verification of Bidirectional Drive Capability***

To verify the bidirectional drive of ITMP, a virtual-controlled real-world test and a real-controlled virtual-world test were conducted. The virtual-controlled real-world test involved constructing a 10m × 10m planar scene using the built-in navigation system in UE4, setting two prominent cylinders as virtual obstacles, and specifying the starting point A(1,1) and the ending point B (8,8). The platform decomposed the path into a series of velocity commands, which were sent to the physical robot controller via ROS for execution, and OptiTrack was used to record the actual motion trajectory of the physical robot. The real-controlled virtual-world test involved manually controlling the physical robot with a handle, randomly executing complex actions including forward, backward, arc movements, and stationary rotations. The movement of the twin in UE4 was observed through screen recording, with 100 samples taken. The verification results of these two tests are given in Table 5. In Table 5, the average deviation of the virtual-controlled real-world test was 0.15m. The radius of the WMR body was approximately 0.34m. In comparison, the WMR was able to avoid error accumulation and collisions with obstacles. These test results demonstrate that UE4 can complete offline path planning and tracking execution, meeting the control accuracy requirements. When encountering a virtual obstacle, WMR successfully executed a detour. Real-world testing results show that the total latency from manual operation to virtual platform update is 162ms. This demonstrates the platform's smoothness and real-time performance.

Table 5: Verification results of bidirectional drive

Virtual control actual testing		Actual control virtual test	
Performance metrics	Numerical results	Delay indicator	Numerical results
Average lateral deviation	0.15 m	Average delay	162 ms
Maximum lateral deviation	0.28 m	Minimum delay	98 ms
Trajectory completion rate	100%	Maximum latency	235 ms
Number of successful obstacle avoidance attempts	2/2	Delay standard deviation	35 ms
Average tracking speed	0.45 m/s		

4. Conclusions and Recommendations

To improve the monitoring accuracy of WMR, an ITMP was constructed, and the GA-BP algorithm was introduced into it for fault diagnosis of the robot drive system. Results showed that the GA-BP algorithm could complete robot fault diagnosis tasks with high accuracy. Compared with the traditional BPNN, GA-BP improved accuracy by 9.5%. Compared with PSO-BP, SVM, and RF, GA-BP maintained good performance in all four metrics. Confusion matrix results confirmed that the GA-BP model had high-precision identification capability for most fault types. The average deviation of the virtual-to-real test was 0.15m, indicating that the constructed ITMP could effectively support virtual-to-real interaction.

The principal achievement of this research was the proposal of an intelligent diagnostic DT monitoring system implementation path, providing an operable technical framework for the digital operation and maintenance of wheeled equipment. However, this study did not consider the collaborative work between multiple WMPs.

In the future, a collaborative architecture could be designed in conjunction with cloud computing to further promote the application of large-scale wheeled equipment.

Fundings

The research is supported by: Project of science and technology research program of Chongqing Education Commission of China (No. KJQN202303910); Chongqing Municipal Education Commission Science and Technology Research Project (No. KJQN202603924).

References

[1] Zhong, R., Hu, B., Hong, Z., Zhang, Z., Lou, S., Song, X., Feng, Y.X., & Tan, J. (2025). Human-robot handover task intention recognition framework by fusing human digital twin and deep domain adaptation. *Journal of Engineering Design*, 36(11), 2055-2071. https://doi.org/10.1080/09544828.2024.2326111

[2] Marinho, M.M., Quiroz-Omaña, J.J., & Harada, K. (2024). A multiarm robotic platform for scientific exploration: Its design, digital twins, and validation. *IEEE Robotics & Automation Magazine*, 31(4): 10-20. https://doi.org/10.1109/MRA.2023.3336472

[3] Yang, G., Ye, Z., Wu, H., Li, C., Wang, R., Kong, D., Hou, Z.Y., Wang, H.F., Huang, X.Y., Pang, Z.B., Dong, N., Pang, G. (2024). A digital twin-based large-area robot skin system for safer human-centered healthcare robots toward healthcare 4.0. *IEEE Transactions on Medical Robotics and Bionics*,

6(3), 1104-1115. https://doi.org/10.1109/TMRB.2024.3421635

[4] Shirakura, N., Maruyama, T., Makihara, K., Ueshiba, T., Itadera, S., Endo, Y., Tada, M., & Domae, Y. (2025). Synergistic effects of robot performance on human–robot mutual assistance systems in manufacturing. *Advanced Robotics*, 39(4), 208-221. https://doi.org/10.1080/01691864.2024.2445600

[5] Li, Y. (2024). Constructing the intelligent expressway traffic monitoring system using the internet of things and inspection robot. *The Journal of Supercomputing*, 80(7), 8742-8766. https://doi.org/10.1007/s11227-023-05794-z

[6] Müller, M., Ruppert, T., Jazdi, N., & Weyrich, M. (2024). Self-improving situation awareness for human–robot-collaboration using intelligent Digital Twin. *Journal of Intelligent Manufacturing*, 35(5), 2045-2063. https://doi.org/10.1007/s10845-023-02138-9

[7] Ye, Z., Jingyu, L., Hongwei, Y. (2025). A digital twin-based human-robot collaborative system for the assembly of complex-shaped architectures. *Proceedings of the Institution of Mechanical Engineers, Part B: Journal of Engineering Manufacture*, 239(12), 1708-1720. https://doi.org/10.1177/09544054221110960

[8] Zhao, L., Nie, Z., Xia, Y., & Li, H. (2024). Virtual-physical tracking control for a car-like mobile robot based on digital twin technology. *IEEE Transactions on Industrial Electronics*, 71(12), 16348-16356. https://doi.org/10.1109/TIE.2024.3384585

[9] Bilal, H., Obaidat, M.S., Aslam, M.S., Zhang, J., Yin, B., & Mahmood, K. (2024). Online fault diagnosis of industrial robot using IoRT and hybrid deep learning techniques: An experimental approach. *IEEE Internet of Things Journal*, 11(19), 31422-31437. https://doi.org/10.1109/JIOT.2024.3418352

[10] Wang, J., Wang, X., Wang, Y., Sun, Y., & Sun, G. (2024). Intelligent joint actuator fault diagnosis for heavy-duty industrial robots. *IEEE Sensors Journal*, 24(9), 15292-15301. https://doi.org/10.1109/JSEN.2024.3377234

[11] Polenghi, A., Cattaneo, L., & Macchi, M. (2024). A framework for fault detection and diagnostics of articulated collaborative robots based on hybrid series modelling of Artificial Intelligence algorithms. *Journal of Intelligent Manufacturing*, 35(5), 1929-1947. https://doi.org/10.1007/s10845-023-02076-6

[12] Zhong, T., Qin, C.J., Shi, G., Zhang, Z., Tao, J., & Liu, C. (2024). A residual denoising and multiscale attention-based weighted domain adaptation network for tunnel boring machine main bearing fault diagnosis. *Science China Technological Sciences*, 67(8), 2594-2618. https://doi.org/10.1007/s11431-024-2734-x

[13] Ramkumar, A., & Balasubramanian, G. (2025). Deep convolutional neural network object net model based cognitive digital twin for trust in human–robot collaborative manufacturing. *Journal of Intelligent Manufacturing*, 36(7), 5141-5161. https://doi.org/10.1007/s10845-024-02501-4

[14] Achour, A., Zaher, M.E., Assaad, H.A., & Dupuis, Y. (2025). Collaborative semantic mapping for updating the digital twin in controlled indoor environment. *Intelligent Service Robotics*, 18(6), 1145-1167. https://doi.org/10.1007/s11370-025-00632-y

[15] Kabir, S.M.H., Rahman, M.T., & Mridul, A.H. (2025). Software defect prediction using traditional machine learning and ensemble learning algorithms. *Smart Wearable Technology*, 1, A9-A9. https://doi.org/10.47852/bonviewSWT52025645

[16] Tipary, B., Erds, G., Kemény, Z. (2025). Constellation-based robotic visual servoing method for fault diagnosis of used printed circuit board assemblies. *CIRP Annals - Manufacturing Technology*, 74(1), 43-47. https://doi.org/10.1016/j.cirp.2025.04.080

[17] Dimitropoulos, N., Kaipis, M., Giartzas, S., & Michalos, G. (2025). Generative AI for automated task modelling and task allocation in human robot collaborative applications. *CIRP Annals - Manufacturing Technology*, 74(1), 7-11. https://doi.org/10.1016/j.cirp.2025.04.081

[18] Zhang, N., Zhao, Y., & Dai, S. (2025). A generic manipulation task instruction framework using interactive digital twin environment. *The International Journal of Advanced Manufacturing Technology*, 137(1), 547-562. https://doi.org/10.1007/s00170-025-15138-1

[19] Walz, D., Sommer, K., Werz, M., & Weihe, S. (2025). Development and Evaluation of a Robot Guided Friction Stir Welding Gun. *International Journal of Precision Engineering and Manufacturing-Green Technology*, 12(6), 1795-1808. https://doi.org/10.1007/s40684-025-00725-0

[20] Wang, L., Wang, Z., Gumma, K., Turner, A., & Ratchev, S. (2025)). Multi-agent cooperative swarm learning for dynamic layout optimisation of reconfigurable robotic assembly cells based on digital twin. *Journal of Intelligent Manufacturing*, 36(5), 2959-2982. https://doi.org/10.1007/s10845-023-02229-7