Design of Intelligent Driving Control Model Combining BP-PID and Deep Attention Mechanism

XiaoJiang QIN

School of Computer Engineering, Chongqing University of Humanities, Science & Technology, Big Data and Information Security Engineering Technology Research Center, Chongqing College of Humanities, Science & Technology, Science & Technology, Hechuan, 401524 Chongqing, China
17300283629@163.com

Abstract: With the acceleration of urbanization, intelligent driving technology has become a key approach for addressing traffic congestion and the increasing number of traffic accidents. It is therefore necessary to explore this technology in depth. However, the existing intelligent driving control methods are still facing limitations with regard to control accuracy, real-time performance, and safety. In order to address these issues, this study proposes an intelligent driving control model that integrates a Convolutional Block Attention Module, a Back Propagation Neural Network, Proportional-Integral-Derivative Control, and multiple optimization algorithms. The experimental results showed that the proposed model achieved a trajectory tracking deviation of 0.18 m, an average steering angle change rate of 3.13°/s during uphill driving, and a slope speed error of 0.46 km/h. Also, the acceleration and deceleration lag time and response time were 1.23 s and 7.2 s, respectively. In addition, the braking trigger delay during cornering was 28 ms, the acceleration fluctuation rate during lane-changing on straight roads was 0.87 m/s^3, and the obstacle-avoidance path offset and lateral acceleration were 0.536 m/s^2 and 0.335 m/s^2, respectively. All these results show that the proposed model outperforms the other employed models, indicating that it effectively improves the overall intelligent driving control performance and provides a valuable reference for the related technologies.

Keywords: Back propagation neural network, Proportional-integral-derivative control, Attention mechanism, Intelligent driving, motion control, Model optimization.

1. Introduction

With the advances in technology and industrial transformation, intelligent driving technology is developing rapidly, and vehicle control plays a key role in determining the reliability of autonomous driving systems (Petrescu et al., 2024; Gao et al, 2023). Intelligent driving requires control models that are accurate, adaptable, and robust. Many researchers have applied intelligent algorithms in areas such as global path planning, environmental perception, and assisted driving. However, the existing methods still have certain limitations. For example, fuzzy predictive control lacks real-time performance in dynamic scenarios, and sliding mode control fails to balance control accuracy and stability. Therefore, there is an urgent need for a more comprehensive intelligent driving control method (Stanojević, B., Dzitac, S. & Dzitac, I., 2020; Bojan-Dragos, C. A., Precup, R. E., Petriu, E. M. et al., 2024). A Back Propagation (BP) Neural Network combined with Proportional-Integral-Derivative (PID) control can adaptively adjust PID control parameters via BP, making it more suitable for nonlinear data (Azegmout et al., 2023). The Convolutional Block Attention Module (CBAM) uses channel and spatial attention for adjusting channel weights and highlighting the key features, thereby improving the representation of local features (Zhang et al., 2023). Based on this, this study combines BP-PID with CBAM to form a hybrid algorithm named CBAM-BP-PID. Meanwhile, Particle Swarm Optimization (PSO), a Deep Q-Network (DQN) and a Pruning Algorithm (PA) were introduced to improve it. As such, this study introduces two main innovations. First, it proposes a CBAM-BP-PID hybrid algorithm that combines BP-PID with CBAM for constructing a vehicle control model. Second, it introduces multiple optimization algorithms to improve it and obtain a final control model, providing a new method for the related research.

The remainder of this paper is structured as follows. Section 2 presents a literature review for the existing studies on BP-PID and the intelligent driving control of vehicles, and summarizes the deficiencies of the previous control models. Section 3 elaborates on the advantages of the methods adopted in this study and explains the construction process and application flow for the final model. Further on, Section 4 analyzes the evaluation results obtained for the final model and compares them with the results in the previous studies. Finally, Section 5 summarizes this paper, it clarifies its shortcomings and outlines future research directions.

2. Literature Review

BP-PID not only inherits the real-time performance and stability of PID control, but it also addresses the limitations of fixed parameters and the poor adaptability to nonlinear data by leveraging BP. Based on these advantages, many scholars have explored this method in depth.

https://doi.org/10.24846/v34i4y202509

For example, to address the low fitting accuracy and long fitting time of PID control in spectral simulation, Yun et al. (2024) put forward a stellar spectrum simulation method based on the BP-PID algorithm. The results showed that, in comparison with PID, the overshoot and response time of BP-PID were reduced by 79.01% and 30%, respectively. Chaturvedi et al. (2024) proposed a novel PSO-tuned PID control for better regulating the temperature of a nonlinear jacketed continuous stirred tank reactor. In comparison with traditional PID controllers, their method proved more effective in suppressing interference signals. In order to address hardware limitations in achieving flight path setting and obstacle avoidance for small uncrewed aerial vehicles (UAVs), Choi & Ha (2024) proposed a wearable controller based on an improved PID controller. This experiment showed that the flight time controlled by this controller for UAVs was reduced from 2.8 s to 1.8 s, which is conducive to improving the application of swarm UAVs. CBAM, with its dual attention mechanism in channel and spatial dimensions, has been widely applied for image classification and intelligent driving perception tasks.

For instance, for addressing the issue of unclear esophageal regions in feature maps generated by the Faster R-CNN backbone, Liu et al. (2023) proposed an improved backbone network based on CBAM to enhance the feature saliency of the esophageal region. The results showed that this method significantly improved the representation of esophageal features. To solve the problems of time consumption and error proneness in manual tea leaf disease detection, Bhuyan et al. (2024) developed a CBAM-based Deep Convolutional Neural Network to improve the recognition accuracy. Their model achieved an average accuracy of 98.27% in identifying tea leaf diseases. Chen, C. et al. (2023) introduced a CBAM-based vehicle detection algorithm built on edge intelligence by enhancing the detection accuracy through channel attention. Results showed that the proposed method improved vehicle detection accuracy from 82.03% to 86.22%.

At present, the research on intelligent driving control has made an inevitable progress, and many scholars have applied it in real-world scenarios. For example, to improve the automatic steering control for autonomous vehicles, Reda et al. (2023) developed three machine learning-based models for autonomous driving tasks. Comparative experiments showed that the Deep Neural Network model effectively controlled autonomous steering. Yang et al. (2023) presented a centralized vehicular network based on a dual DAO architecture for ensuring data accuracy in intelligent driving. By distinguishing between real-time and non-real-time decision tasks, their system performed data validation and proved effective in maintaining data security, integrity, and accuracy. For addressing the limitations of traditional obstacle-avoidance trajectory planning in ensuring vehicle safety and comfort, Li et al. (2023) proposed an online collision-avoidance trajectory planning method for optimizing the longitudinal and lateral trajectories. Results indicated that this method significantly enhanced the driving safety and comfort. Zhang et al. (2024) proposed a cloud-based collaborative architecture supporting autonomous-driving Artificial Intelligence-Generated Content (AIGC) to improve vehicle perception and decision-making. Their approach used AIGC for supporting system design and resource management, demonstrating enhanced capabilities in motion planning and trajectory prediction. Further on, to address unreliable communication links in wireless distributed consensus networks, Feng et al. (2023) proposed a three-phase consensus mechanism based on the practical Byzantine fault-tolerant algorithm. Their study showed that this mechanism could still enable data transmission under poor wireless communication conditions. Hu & Huang (2024) proposed an autonomous driving decision-making model based on an improved actor-critic algorithm to enhance the autonomous driving systems' ability to handle complex, variable traffic scenarios. The results show that this model can better handle complex, variable traffic scenarios and ensure the safety of pedestrians and drivers.

In summary, although the current research in intelligent vehicle control has achieved considerable results, the existing intelligent driving control methods for vehicles are still facing problems such as being easily disturbed by redundant information and having a relatively high computational complexity. Therefore, this study proposes a model based on the CBAM-BP-PID algorithm, further optimized by PSO, DQN, and PA, to address the problems of low accuracy, poor real-time performance and stability, and high computational cost inherent to the current control methods.

3. Research Methodology

3.1 Intelligent Driving Control Model Nased on BP-PID and CBAM

Intelligent driving control refers to the use of intelligent algorithms to perceive environmental information and generate real-time control signals to enable intelligent vehicle driving (Chen, L. et al., 2023; Deng et al., 2023). BP-PID dynamically optimises the PID control parameters by leveraging self-learning for BP, effectively overcoming the limitations of manual parameter tuning and the poor adaptability of traditional PID (Lin, Jhang & Chuang, 2024). Therefore, this study applies BP-PID to vehicle intelligent driving control. Its specific structure is shown in Figure 1.

As shown in Figure 1, during intelligent driving control using BP-PID, the vehicle's current state, such as speed and acceleration, is first processed by the BP network. The input and hidden layers process it, and the output layers establishes a mapping to the core PID parameters. Next, the network weights are dynamically adjusted based on the parameter mapping error to adapt to changes in vehicle states, and the optimal PID parameters are obtained and sent to the controller to complete the intelligent driving control. The core equation of PID output is shown in equation (1):

$$u(t) = K_p e(t) + K_i \int_0^t e(\tau) d\tau + K_d \frac{d_e(t)}{dt} \tag{1}$$

Here, $u(t)$ defines the PID output at time t, $e(t)$ is the system error, K_p is the proportional coefficient, K_i is the integral coefficient, and K_d is the derivative coefficient. The update process for the BP output layer is shown in equation (2):

$$\Delta\omega_{lj} = \eta \cdot \frac{\partial E}{\partial \omega_{lj}} + \alpha \cdot \Delta\omega_{lj}(k-1) \tag{2}$$

In equation (2), $\Delta\omega_{lj}$ represents the weight update, j and l are the nodes of the hidden and output layer, respectively, ω_{lj} defines the output layer weight, η is the learning rate, α is the momentum factor, and k is the number of updates. However, BP-PID has a limited capability to process vehicle features under complex road conditions and shows a poor adaptability to highly nonlinear problems. In order to address this problem, CBAM is introduced for its robust feature extraction capabilities. CBAM captures features of nonlinear problems through its attention mechanism and improves both feature representation and adaptability to strong nonlinearity (Liu & Zhou, 2024). Therefore, this study combines CBAM with BP-PID to form the CBAM-BP-PID hybrid algorithm, which helps control the vehicle parameters more accurately and improve control precision. The structure of the hybrid algorithm is shown in Figure 2.

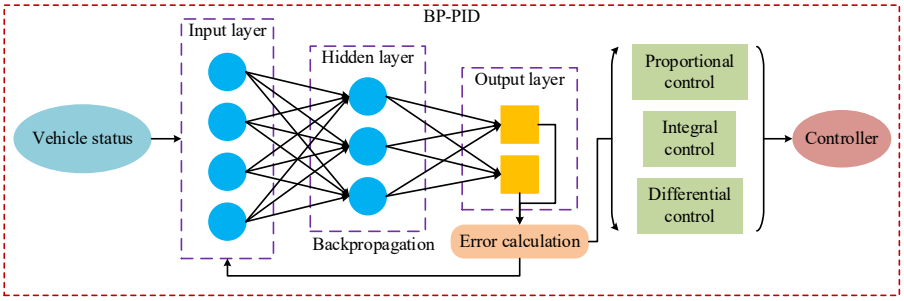


Figure 1. BP-PID structure for intelligent driving control

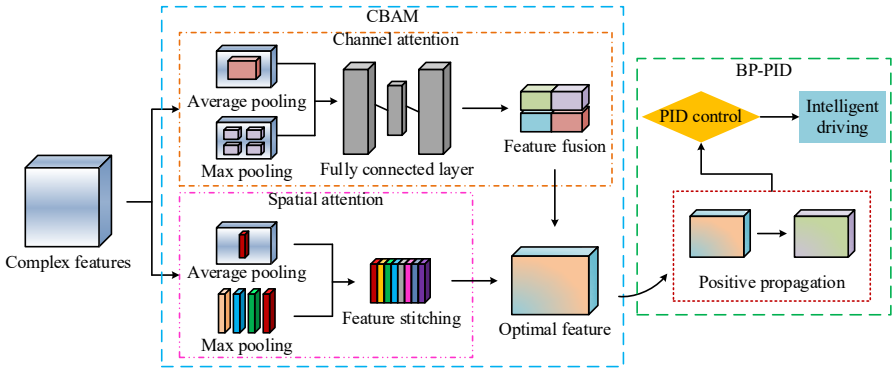


Figure 2. Structure of the CBAM-BP-PID hybrid algorithm

As shown in Figure 2, the intelligent driving control process includes CBAM preprocessing of the sensed data features. The channel attention module analyses the importance of each channel for highlighting those which are the most relevant to driving decisions, while the spatial attention module focuses on key regions in space. Together, they generate optimal features. BP then dynamically adjusts the PID control parameters based on the features extracted by CBAM, thereby enabling an intelligent vehicle control. The pooling operation in the channel attention module of CBAM is shown in equation (3) (Ding et al., 2024):

$$M_c(F) = \sigma \cdot (\mathrm{MLP}(AvgPool(F)) + \mathrm{MLP}(MaxPool(F))) \quad (3)$$

In equation (3), $M_C(F)$ is the pooling operation for the channel attention module, F is the input feature map, $AvgPool(F)$ represents the global average pooling, $MaxPool(F)$ is the global max pooling, MLP is a multilayer perceptron, and σ is the Sigmoid function. The pooling operation function of the CBAM spatial attention module is expressed in equation (4):

$$M_s(F) = \sigma \cdot (f^{7\times7}(|AvgPool(F); MaxPool(F)|)) \quad (4)$$

In equation (4), $M_s(F)$ is the pooling operation for the spatial attention module, and f is the convolution operation. In comparison with the traditional control methods, the CBAM-BP-PID algorithm combines the collaborative control capabilities of the attention mechanism and PID. It uses BP to optimize the PID parameters in real time, thereby helping to ensure the safety and reliability of autonomous driving. Based on this, this study proposes an intelligent driving control model using CBAM-BP-PID. The specific process is shown in Figure 3.

As shown in Figure 3, the control process for this model includes four steps. First, the system perceives the vehicle environment and status. CBAM then filters the redundant information and extracts the key features. Further on, BP dynamically updates the feature parameters, and PID generates control commands to drive the vehicle. Finally, the feedback analysis module calculates the deviations from the driving results and updates the control instructions, enabling a real-time intelligent driving control. BP usually optimizes weights by minimizing the vehicle control error. The error function is expressed in Equation (5):

$$E = \frac{1}{2}\left[r(k') - y(k')\right]^2 \quad (5)$$

In equation (5), E defines the error function, k' represents weight optimization, and $r(k') - y(k')$ is the system error. After CBAM is introduced, the input to the BP hidden layer turns into the weighted features. The specific process is shown in Equation (6):

$$x_i^{'} = F_i^{''} \quad i = 1, 2, 3, \cdots \quad (6)$$

In Equation (6), $x_i^{'}$ represents the feature of the i-th node in the input layer after weighting, and F'' is the CBAM-optimized feature map.

3.2 Optimization of the Intelligent Driving Control Model for Different Scenarios

Although the CBAM-BP-PID-based control model demonstrates a strong intelligent driving control performance, it still faces problems such as a poor coordination between decision-making and control, a tendency to fall into local optima, and a high computational complexity. Therefore, it needs further improvement. PSO performs the global optimization of parameters via particle iteration and a fitness function, thereby reducing the likelihood of the model falling into local optima (Liu, Liu & Gong, 2021). This study applies PSO for optimizing the initial weights and parameters of BP, aiming to reduce the likelihood of local minima and better align parameter optimization with the global control objectives. The optimization process is shown in Figure 4.

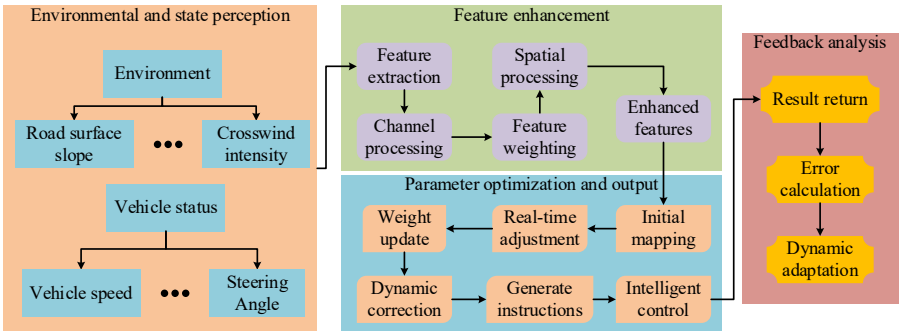


Figure 3. Process of the CBAM-BP-PID-based control model

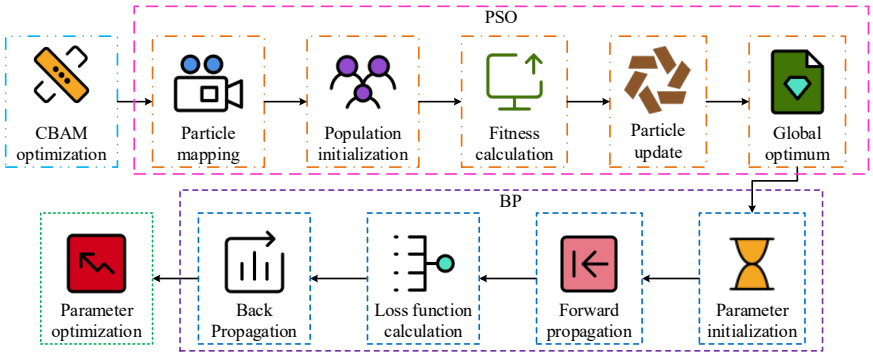


Figure 4. Schematic diagram of the process of PSO optimization for the BP network (Anon, n.d.)

As shown in Figure 4, when PSO improves BP, it first encodes the CBAM-processed features into particles and initializes the population. Then it calculates the fitness values, updates the particle velocity and position, determines the individual and global optima, and uses them as the initial weights for BP. Finally, BP completes an iterative training via forward propagation and loss function calculation, converging to the global optimal parameters and thereby avoiding the local optima. The fitness function of PSO is expressed in equation (7) (Minh et al., 2023):

$$fitness = \sum_{k^*=1}^{N} \left[r(k^*) - y(k^*) \right]^2 \quad (7)$$

In equation (7), *fitness* is the fitness function, N is the number of sampling points, $r(k^*)$ is the target value, and $y(k^*)$ is the actual system output. After combining PSO and BP, the process of updating particle velocity and position to find the global optimum is shown in equation (8):

$$g_{best} = \arg\min_{i^*}^{N^*} fitness(p_{best_{i^*}}) \quad (8)$$

In equation (8), g_{best} is the global best particle, $p_{best_{i^*}}$ defines the individual best particle, and N^* is the number of particles. DQN, as an algorithm that integrates deep learning and reinforcement learning, improves model stability by fitting the Q-function and setting action rewards (Bayazit, Sahingoz & Dogan, 2023). Based on this, this study employs DQN to enhance the model's environmental perception, enabling it to quickly optimize decision-making logic in response to dynamic environmental changes and achieve a real-time coordination between decision-making and control. The state space function of DQN is shown in equation (9):

$$S = \left[e(o), ec(o), K_p(o), K_i(o), K_d(o), CBAM_{att}(o) \right] \quad (9)$$

In equation (9), S represents the state vector, $e(o)$ is the current error, $ec(o)$is the error change rate, $K_p(o)$, $K_i(o)$, and $K_d(o)$ are the PID parameters, and $CBAM_{att}(o)$ is the attention weight for CBAM. The reward function is expressed in equation (10):

$$R(S, A) = w_1 \cdot e - \left| e(k'') \right| - w_2 \cdot \left| \Delta\mu(k'') \right| - w_3 \cdot \left| \Delta A \right| \quad (10)$$

In equation (10), $R(S,A)$ is the reward function, w_1, w_2, and w_3 are weight coefficients, $\Delta\mu(k'')$ is the change in the control variable k'', and ΔA is the amplitude of the action. PA can optimize the model structure by using structured pruning strategies, thereby reducing the computational complexity while maintaining the original performance (Wei, Yan & Yan, 2024). Therefore, this study introduces PA for simplifying the proposed model, thereby reducing the number of parameters and storage requirements and improving the overall efficiency. The improvement process is shown in Figure 5.

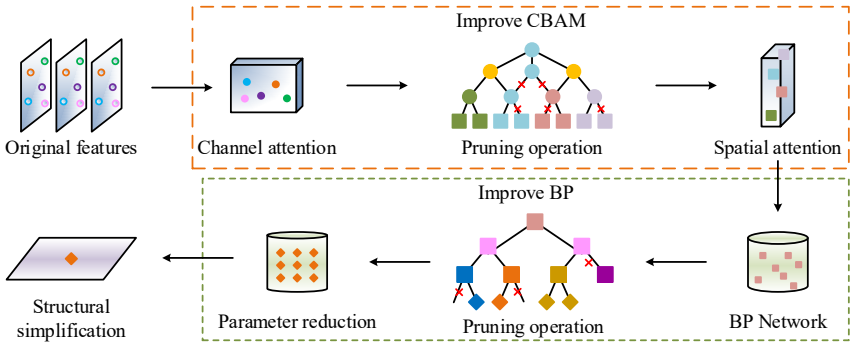


Figure 5. Schematic diagram of the process of PA optimization for the proposed model

As shown in Figure 5, PA mainly improves the channel and spatial attention modules of CBAM and the connections in BP. After receiving the raw features, PA reduces the redundant weight computations in CBAM via pruning, thereby accelerating the selection of key features. During the BP parameter optimization, it removes the neurons or connections with a low contribution through further pruning, reducing the number of parameters and simplifying the model structure. The pruning process of CBAM's channel attention module is shown in equation (11):

$$\tilde{M}_c(F)_{j^*} = \begin{cases} M_c(F)_{j^*}, & \text{if } \sum_{m',n'} \left| F_{j^*,m',n'} \right| \geq \varsigma \\ 0, & \text{otherwise} \end{cases} \quad (11)$$

In Equation (11), $\tilde{M}_c(F)_{j^*}$ is the pruning operation, $M_c(F)_{j^*}$ is the pooling operation, ς represents the minimum contribution, $F_{j^*,m',n'}$ is defined as the input feature map of the dimension (j^*, m', n'). The weight pruning function applied to BP by the PA is shown in equation (12) (Huang et al., 2023):

$$\tilde{W}_{ij} = \begin{cases} W_{ij}, & \text{if } \left| W_{ij} \right| \geq \varsigma \\ 0, & \text{otherwise} \end{cases} \quad (12)$$

In Equation (12), $\tilde{W}_{ij}$ represents weight pruning and W_{ij} denotes BP parameter optimization. This study applies PSO, DQN, and PA for improving the control model in a targeted manner, addressing issues such as the susceptibility to local optima, the poor coordination in dynamic decision control, and the high computational complexity. Based on this, this study constructs an intelligent driving control model that combines CBAM-BP-PID with multiple optimization algorithms, being referred to as the final model. The control process of the final model is shown in Figure 6.

As shown in Figure 6, the control process of the final model is as follows. After the sensors collect the data on road conditions, environmental data, and vehicle parameters, CBAM enhances the key features and sends them to BP. PSO then optimizes BP's initial weights and thresholds. The parameters are passed to PID to generate preliminary control commands. At the same time, DQN receives these preliminary commands as input and dynamically adjusts them using a reward function for achieving decision-control coordination. Finally, PA prunes the model to simplify its structure and outputs a lightweight intelligent driving instruction. The overall output mapping for the final model is expressed in equation (13):

$$u^*(t) = f^*(x^*(t); \theta_{PA}, \theta_{DQN}, \theta_{PSO}, \theta_{CBAM}) \quad (13)$$

In equation (13), $u^*(t)$ is the output control at time t, $f^*(\cdot)$ defines the mapping relation, $x^*(t)$ is the data on input perception, θ_{PA} is the pruned sparse parameter, θ_{DQN} is the reinforcement learning decision parameter, θ_{PSO} is the optimized BP-PID parameter, and θ_{CBAM} is the CBAM attention weight.

4. Results and Discussion

4.1 Accuracy and Stability Evaluation for the Final Model

To verify the feasibility of the final model, this study compared it with the Proximal Policy Optimization and Convolutional Neural Network (PPO-CNN) model, the Improved Safety Zone (ISZ) model, and the Improved Fuzzy Safety (IFS) model. A BYD passenger vehicle was selected as the test object, and the effectiveness of autonomous driving under the control of the four employed models was compared. In the experiment, a central city road with a speed limit of 60 km/h was selected, and a real driving trajectory was predefined. Each model then controlled the vehicle in the context of autonomous driving. The

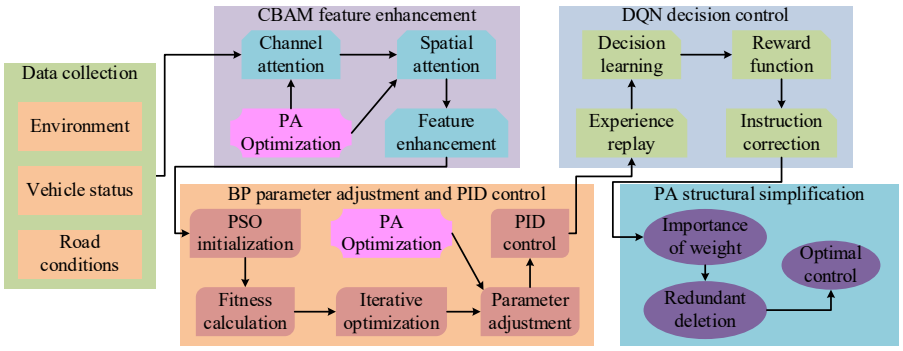


Figure 6. Control process of the final model for intelligent driving

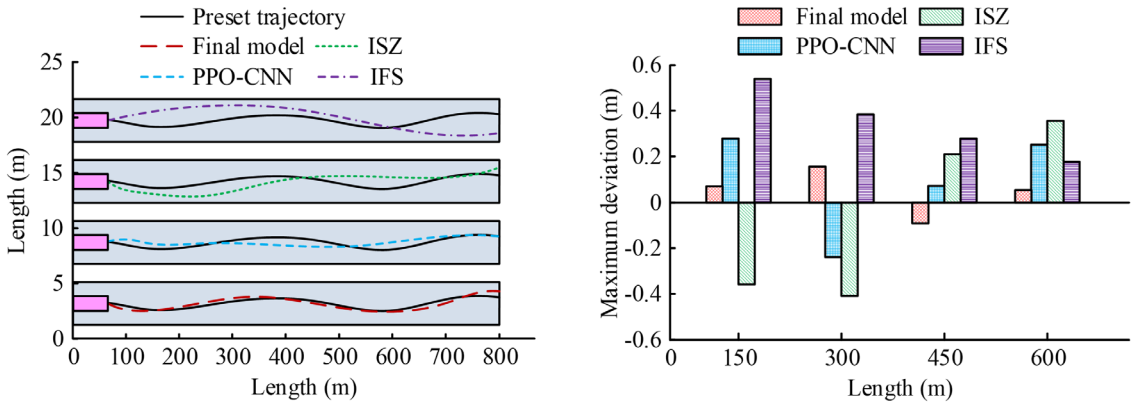
driving trajectories and their deviations from the preset trajectory were recorded. The study first compared the driving trajectories and then the trajectory deviations under the control of the four models. The results are shown in Figure 7.

As shown in Figure 7(a), the driving trajectory generated by the final model was the closest to the preset path, with almost no visible deviation. In contrast, all the other models showed varying degrees of deviation. Figure 7(b) shows that the final model reached a maximum trajectory deviation of 0.18 m at the 300 m mark, significantly better than 0.28 m for the PPO-CNN model, 0.41 m for the ISZ model, and 0.53 m for the IFS model. These results indicated that the final model had a higher control accuracy. This was because the DQN optimization enhanced the coordination of dynamic decision-making control, enabling the real-time correction of minor trajectory deviations and the timely adjustment of control strategies to reduce errors. The study then compared the four models in terms of steering angle variation rate, slope speed error, and acceleration-deceleration delay time during uphill driving. The results are shown in Table 1.

Based on Table 1, the final model achieved an average steering angle variation of 3.13°/s, an average slope speed error of 0.46 km/h, and an average acceleration-deceleration delay time of 1.23 s, outperforming the other three models. These results showed that the final model allowed a smoother steering control and featured a greater precision, with a stronger stability when facing resistance. The low variation rate for the steering angle was due to the CBAM module, which enhanced the perception of key features such as lane lines and slope gradients. This provided a more reliable input to the BP-PID module, effectively reducing the need for frequent steering adjustments. In summary, the final model demonstrated a higher control accuracy and stability in the context of autonomous driving, supporting its engineering implementation for urban roads and highways.

4.2 Real-time Performance Evaluation for the Final Model

After verifying the accuracy and stability of the final model, this study evaluated its real-time performance. To ensure reliability, the test vehicle



(a) Comparison of the trajectories for autonomous driving of the vehicle controlled by the four models

(b) Comparison of the trajectory deviations for autonomous driving of the vehicle controlled by the four models

Figure 7. Comparison of the driving trajectories and trajectory deviations under the four employed models

Table 1. Comparison of three indicators when the vehicle is going uphill

Indicator	Number of experiments	Model			
		Final model	PPO-CNN	ISZ	IFS
Steering angle variation (°/s)	10	3.6	7.3	6.8	8.4
	20	3.0	6.7	6.4	7.9
	30	2.8	6.6	6.0	7.5
Slope speed error (km/h)	10	+0.73	-1.81	+2.36	+2.12
	20	+0.36	-1.52	+2.04	+1.75
	30	+0.29	-1.40	+1.88	+1.59
Acceleration-deceleration delay time (s)	10	1.8	2.6	4.0	3.3
	20	1.3	2.4	3.7	3.1
	30	0.6	2.0	3.5	2.8

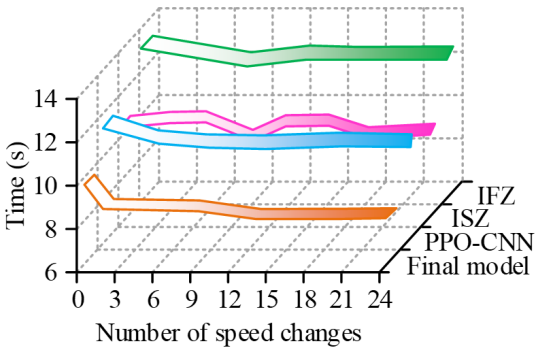
and model remained unchanged. A straight highway segment with a 120 km/h speed limit was selected. The required time and resource usage for acceleration and deceleration in the 0–100 km/h range were recorded. The time consumption and resource usage during lane changes are shown in Figure 8.

As shown in Figure 8(a), the time required for speed changes under the final model gradually stabilized at the twelth attempt, reaching a minimum of 7.2 s. This was significantly shorter than 11.6 s for PPO-CNN, 10.4 s for ISZ, and 12.7 s for IFS. Figure 8(b) shows that the resource usage during lane changes under the final model remained lower than that of the other three models. It stabilized around the 36-minute mark, with an average resource usage rate as low as 39.6%, outperforming all the other models. Subsequently, on a closed test road with straight and continuously curved segments, this study compared the braking trigger delay and acceleration fluctuation during lane changes for the four models. The results are presented in Table 2.

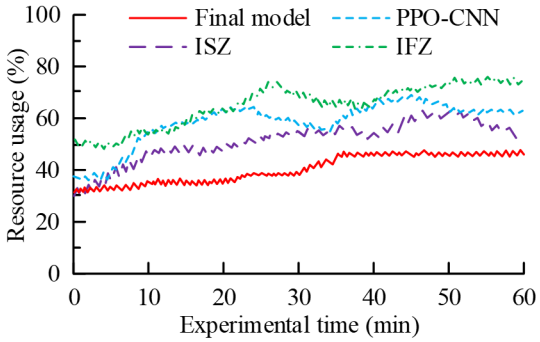
As shown in Table 2, the braking delay under the final model reached a minimum of 28 ms after 40 tests and the acceleration fluctuation during straight-line lane changes reached a minimum of 0.87 m/s^3 at the 30th test and remained stable afterwards. These results were better than those obtained by the other three models. This performance was attributed to the PA module, which reduced the computational delay through a streamlined architecture and leveraged DQN's predictive ability to mitigate cornering risks, thereby shortening the braking trigger delay. Meanwhile, the BP-PID module ensured a smoother steering and acceleration-deceleration control, resulting in a lower acceleration fluctuation rate during lane changes. In conclusion, the final model enabled a faster response in acceleration, deceleration, and cornering scenarios, with lower resource usage requirements, making it suitable for an efficient and reliable real-time control under full driving conditions.

4.3 Safety Evaluation for the Final Model

After verifying the real-time performance of the final model, this study evaluated its safety. To ensure reliability, the test vehicle and model remained the



(a) Comparison of the time consume for vehicle acceleration and deceleration controlled by each model



(b) Comparison of resource occupancy rates when each model controls vehicle lane changes

Figure 8. Comparison of the acceleration-deceleration time and lane-change resource usage for the four models

Table 2. Comparison of the braking delay for curves and acceleration fluctuation during straight-line lane changes

Indicator	Number of experiments	Model			
		Final model	PPO-CNN	ISZ	IFS
Braking delay (ms)	10	72	124	151	140
	20	55	102	134	128
	30	31	96	117	114
	40	28	90	105	109
	50	28	83	96	100
Acceleration fluctuation (m/s^3)	10	+1.46	+2.20	-3.01	+2.59
	20	+1.12	+2.03	-2.75	+2.36
	30	+0.87	+1.86	-2.33	+2.24
	40	+0.87	+1.64	-2.16	+2.08
	50	+0.87	+1.64	-1.95	+1.86

same. On a closed test road obstacles were placed, and the four models were tested with regard to their obstacle-avoidance performance. The deviation from the planned path and the lateral acceleration were recorded. The results are shown in Figure 9.

As shown in Figure 9, the minimum path deviation under the final model during obstacle avoidance was 0.142 m. This increased gradually to 0.536 m and dropped to 0.117 m at the 500 m mark. After that, the vehicle continued without deviation. In contrast, the other three models exhibited deviations even during everyday driving, and their recovery distances after obstacle avoidance were significantly longer. Table 1 also shows that the average lateral acceleration under the final model was 0.335 m/s^2, much lower than that of the other models, and it was only observed between the 150 m and 550 m marks. Finally, a car-following simulation was conducted in order to compare the braking distance after triggering deceleration. The results are shown in Figure 10.

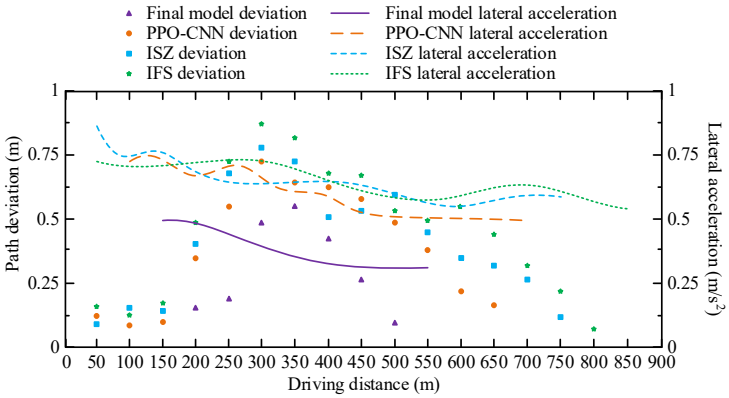
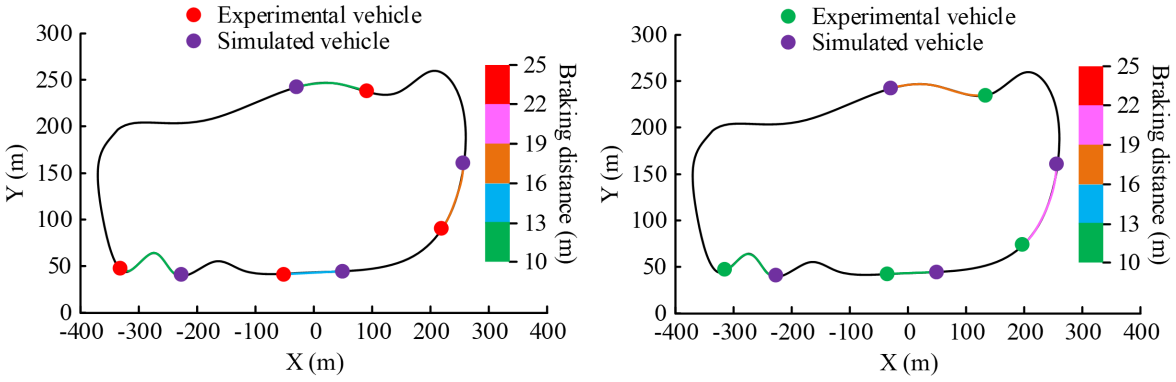


Figure 9. Comparison of path deviation and lateral acceleration during obstacle avoidance for the four models



(a) Final model controls the car-following braking distance for the vehicle's intelligent driving

(b) PPO-CNN controls the car-following braking distance for the vehicle's intelligent driving

(c) ISZ controls the car-following braking distance for the vehicle's intelligent driving

(d) IFS controls the car-following braking distance for the vehicle's intelligent driving

Figure 10. Comparison of braking distance during the car-following scenario for the four models

As shown in Figure 10(a), the vehicle under the control of the final model had a braking distance of 13–16 m on straight roads, of 10–13 m during wide turns, and of 16–19 m for continuous curves. These distances were moderate and reasonable, ensuring safety during car-following. Figures 10(b) to 10(d) show that, for continuous curves, the braking distances for the PPO-CNN, ISZ and IFS models were shorter than that of the final model. During wide turns, both PPO-CNN and IFS had significantly longer braking distances than the final model (the exception being ISZ), and on straight roads, both ISZ and IFS had longer braking distances than the final model (the exception being PPO-CNN. In summary, the final model demonstrated a greater safety during obstacle-avoidance and car-following scenarios. It effectively reduced the collision risk and improved the reliability of intelligent driving systems in complex traffic environments.

4.4 Robustness and Computational Efficiency Evaluation for the Final Model

At last, in order to verify the adaptability and operational efficiency of the final model under complex working conditions, this study assessed the model's robustness by simulating interference from extreme environments. The computational efficiency of the model was evaluated by quantifying the consumption of computing resources and the reasoning time. The experimental subjects were the same, namely BYD household vehicles, and the comparison models were the same, that is PO-CNN, ISZ and IFS to ensure the reliability of the data. As the robustness test isconcerned, three typical interference scenarios were selected. The first consist in adding a crosswind interference of ±5 m/s in the urban road experiment to simulate the impact of bad weather on vehicle driving. The second consists in implementing a sudden change in the road surface friction coefficient in high-speed sections to simulate a sudden change in road conditions. The third one involves adding Gaussian noise to the sensor data to simulate the hardware's perception error. With regard to the computing efficiency test, it is mainly carried out along two dimensions: real-time power consumption and the time required for a single calculation. The test for real-time computing power consumption was conducted on a vehicle-mounted embedded chip as the experimental platform, and the average computing power required by each model to process environmental data was recorded. The specific experimental results are included in Table 3.

Table 3 shows that the maximum trajectory deviation for the final model increased by only 0.06 m under crosswind interference. When the road surface friction coefficient suddenly changed, the steering wheel angle adjustment delay increased by only 4.1 ms, and no oversteering or understeering occurred. Under the interference of Gaussian noise, the fluctuation value for the speed control error of the final model is only 0.13 km/h. In contrast, for the other models under the same interference conditions, the increase in trajectory deviation exceeded 0.15 m, the range of steering wheel angle adjustment delay increased to 8-12 ms, and the fluctuation range for the speed control error reached 0.25-0.35 km/h. This indicates that the final model, which leverages CBAM's feature screening and PSO's parameter optimization, can effectively filter out crosswind interference while maintaining the vehicle control stability. Furthermore, Table 3 shows that the average computing power consumption for the final model is 18.2GB, which is significantly lower than 35.9GB for the PPO-CNN model, 42.0GB for the ISZ model, and 38.7GB for the IFS model. As regards the time consumption for a single calculation, after PA pruned the final model, its time consumption for

Table 3. Comparison of the robustness and computational efficiency between the final model and the other employed models

Evaluation dimension	Indicator	Experimental model			
		Final model	PPO-CNN	ISZ	IFS
Robustness	Trajectory deviation (m)	0.06	>0.15	>0.15	>0.15
	The steering wheel angle adjustment delay (ms)	4.1	8-12	8-15	10-15
	The speed control error fluctuation (km/h)	0.13	0.24	0.30	0.35
Computational efficiency	Average computing power consumption (GB)	18.2	35.9	42.0	38.7
	Duration of a single calculation (ms)	81.4	143.9	194.8	235.1

calculating a single vehicle control instruction was as low as 81.4 ms, while the PPO-CNN model, which had the second lowest calculation time consumption in comparison with the ISZ and IFS models still required 143.9 ms to complete the inference calculation. The above data shows that the structured pruning of the PA algorithm can significantly reduce the computational complexity and hardware resource requirements of the final model while retaining its performance, thereby improving its computational efficiency.

5. Conclusion

In order to address the issues of low accuracy, poor real-time performance, and limited safety of the existing intelligent driving control methods, this study combined BP-PID with the CBAM to form a fusion algorithm. Based on this, an intelligent driving control model was constructed. Furthermore, PSO, DQN, and PA were employed for optimizing the CBAM-BP-PID model, resulting in a final control model. The experimental results showed that the final model achieved a maximum trajectory tracking deviation of 0.18 m. During uphill driving, the average steering angle variation rate was 3.13°/s, the speed error on slopes was 0.46 km/h, and the acceleration-deceleration delay time was 1.23 s. In addition, the minimum time consumption for speed variation and the average resource usage rate were 7.2 s and 39.6%, respectively. The braking delay during cornering reached 28 ms, the acceleration fluctuation rate during straight-line lane changes reached a minimum of 0.87 m/s^3, and the path deviation during obstacle avoidance and the average lateral acceleration were 0.536 m and 0.335 m/s^2, respectively. The values of all these performance indicators were superior to those obtained by the other three models. At the same time, this study also verified the robustness and computational efficiency of the final model, and its performance was also superior to that of the other three models. Anyhow, although the final model demonstrated a strong performance, this study did not explore its robustness and generalization capability in depth. Therefore, future research should focus on these aspects in order to further enhance the reliability of intelligent driving control systems.

Acknowledgements

This research was supported by the Curriculum Reform and Practice of "Principles and Applications of Database Systems" in Application-Oriented Universities under the Background of New Engineering Disciplines (Project No. 22CRKXJJGZX19).

REFERENCES

Anon. (n.d.) *IconPark Homepage* https://iconpark.oceanengine.com/home [Accessed 25th July 2025].

Azegmout, M., Mjahed, M., El Kari, A. et al. (2023) New Meta-heuristic-Based Approach for Identification and Control of Stable and Unstable Systems. *International Journal of Computers, Communications & Control*. 18(4), Art. ID 5294. https://doi.org/10.15837/ijccc.2023.4.5294.

Bayazit, C. E., Sahingoz, O. K. & Dogan, B. (2023) Deep Learning based Malware Detection for Android Systems: A Comparative Analysis. *Tehnički vjesnik* [*Technical Gazette*], 30(3), 787-796. https://doi.org/10.17559/TV-20220907113227.

Bhuyan, P., Singh, P. K. & Das, S. K. (2024) Res4net-CBAM: A deep cnn with convolution block attention module for tea leaf disease diagnosis. *Multimedia Tools and Applications*. 83(16), 48925-48947. https://doi.org/10.1007/s11042-023-17472-6.

Bojan-Dragos, C. A., Precup, R. E., Petriu, E. M. et al. (2024) Sliding Mode and Super-Twisting Sliding Mode Control Structures for Vertical Three-Tank Systems. *Studies in Informatics and Control*. 33(3), 5-16. https://doi.org/10.24846/v33i3y202401.

Chaturvedi, S., Kumar, N. & Kumar, R. (2024) A PSO-optimized novel PID neural network model for temperature control of jacketed CSTR: design, simulation, and a comparative study. *Soft Computing*. 28(6), 4759-4773. https://doi.org/10.1007/s00500-023-09138-0.

Chen, C., Wang, C., Liu, B. et al. (2023) Edge Intelligence Empowered Vehicle Detection and Image Segmentation for Autonomous Vehicles. *IEEE Transactions on Intelligent Transportation Systems*. 24(11), 13023-13034. https://doi.org/1109/TITS.2022.3232153.

Chen, L., Li, Y., Huang, C. et al. (2023) Milestones in Autonomous Driving and Intelligent Vehicles - Part I: Control, Computing System Design, Communication, HD Map, Testing, and Human Behaviors. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*. 53(9), 5831-5847. https://doi.org/10.1109/TSMC.2023.3276218.

Choi J-H & Ha O-K. (2024) Enhancing Microdrone Control: P-PID Controller and Wearable Interface. *Studies in Informatics and Control*. 33(3), 83-92. https://doi.org/10.24846/v33i3y202408.

Deng, B., Nan, J., Cao, W. et al. (2023) A Survey on Integration of Network Communication into Vehicle Real-Time Motion Control. *IEEE Communications Surveys & Tutorials*. 25(4), 2755-2790. https://doi.org/10.1109/COMST.2023.3295384.

Ding, L., Wang, L., Wang, Y. et al. (2024) A Text Recognition Algorithm Based on a Dual-Attention Mechanism in Complex Driving Environment. *Tehnički vjesnik* [*Technical Gazette*]. 31(1), 247-253. https://doi.org/10.17559/TV-20231023001052.

Feng, C., Xu, Z., Zhu, X. et al. (2023) Wireless Distributed Consensus in Vehicle to Vehicle Networks for Autonomous Driving. *IEEE Transactions on Vehicular Technology*. 72(6), 8061-8073. https://doi.org/10.1109/TVT.2023.3243995.

Gao, Y., Tong, W., Wu, E. Q. et al. (2023) Chat with ChatGPT on Interactive Engines for Intelligent Driving. *IEEE Transactions on Intelligent Vehicles*. 8(3), 2034-2036. https://doi.org/10.1109/TIV.2023.3252571.

Hu R. & Huang P. (2024) Autonomous Driving Decision-Making Based on an Improved Actor-Critic Algorithm. *Studies in Informatics and Control*. 33(4), 37-50. https://doi.org/10.24846/v33i4y202404.

Huang, S., Li, D., Zhang, Z. et al. (2023) CSLSEP: an ensemble pruning algorithm based on clustering soft label and sorting for facial expression recognition. *Multimedia Systems*. 29(3), 1463-1479. https://doi.org/10.1007/s00530-023-01062-5.

Li, G., Zhang, X., Guo, H. et al. (2023) Real-Time Optimal Trajectory Planning for Autonomous Driving with Collision Avoidance Using Convex Optimization. *Automotive Innovation*. 6(3), 481-491. https://doi.org/10.1007/s42154-023-00222-7.

Lin, C.-J., Jhang, J.-Y., & Chuang, C.-C. (2024) Navigation Control of an Autonomous Ackerman Robot in Unknown Environments by Using a Lidar-Sensing-Based Fuzzy Controller. *Computer Science and Information Systems*. 21(2), 473-490. https://doi.org/10.2298/CSIS220826008L.

Liu, J.-Y., Liu, S.-F. & Gong, D.-Q. (2021) Electric Vehicle Charging Station Layout Based on Particle Swarm Simulation. *International Journal of Simulation Modelling*. 20(4), 754-765. https://doi.org/10.2507/IJSIMM20-4-CO17.

Liu, B., Zhao, X., Hu, H. et al. (2023) Detection of Esophageal Cancer Lesions Based on CBAM Faster R-CNN. *Journal of Theory and Practice of Engineering Science*. 3(12), 36-42. https://doi.org/10.53469/jtpes.2023.03(12).06.

Liu, J. & Zhou, G. (2024) Learning Discriminative Representations through an Attention Mechanism for Image-based Person Re-identification. *Computer Science and Information Systems*. 21(4), 1483-1498. https://doi.org/10.2298/CSIS230829044L.

Minh, H.-L., Khatir, S., Rao, R. V. et al. (2023) A variable velocity strategy particle swarm optimization algorithm (VVS-PSO) for damage assessment in structures. *Engineering with Computers*. 39(2), 1055-1084. https://doi.org/10.1007/s00366-021-01451-2.

Petrescu, M., Ștefan, M. C., Panagoreț, A. A. et al. (2024) Route Planning and Machine Learning Algorithms for Sensor-Equipped Autonomous Vehicles in Agriculture. *Studies in Informatics and Control*. 33(4), 105-112. https://doi.org/10.24846/v33i4y202410.

Reda, A., Benotsmane, R., Bouzid, A. A. et al. (2023) A Hybrid Machine Learning-based Control Strategy for Autonomous Driving Optimization. *Acta Polytechnica Hungarica*. 20(9), 165-186. https://doi.org/10.12700/aph.20.9.2023.9.10.

Stanojević, B., Dzitac, S. & Dzitac, I. (2020) Fuzzy numbers and fractional programming in making decisions. *International Journal of Information Technology & Decision Making*. 19(4), 1123-1147. https://doi.org/10.1142/S0219622020300037.

Wei, Z. H., Yan, L. & Yan, X. (2024) Optimizing Production with Deep Reinforcement Learning. *International Journal of Simulation Modelling (IJSIMM)*. 23(4), 692-703. https://doi.org/10.2507/IJSIMM23-4-CO17.

Yang, J., Ni, Q., Luo, G. et al. (2023) A Trustworthy Internet of Vehicles: The DAO to Safe, Secure, and Collaborative Autonomous Driving. *IEEE Transactions on Intelligent Vehicles*. 8(12), 4678-4681. https://doi.org/10.1109/TIV.2023.3337345.

Yun, Z., Zhang, Y., Liu, Q. et al. (2024) Research on the simulation method of a BP neural network PID control for stellar spectrum. *Optics Express*. 32(22), 38879-38895. https://doi.org/10.1364/OE.536964.

Zhang, X., Cao, X., Zhang, H. et al. (2023) An Intelligent Obstacle Detection for Autonomous Mining Transportation with Electric Locomotive via Cellular Vehicle-to-Everything and Vehicular Edge Computing. *IEEE Transactions on Intelligent Transportation Systems*. 25(3), 3177-3190. https://doi.org/1109/TITS.2023.3324145.

Zhang, J., Wei, Z., Liu, B. et al. (2024) Cloud-Edge-Terminal Collaborative AIGC for Autonomous Driving. *IEEE Wireless Communications*. 31(4), 40-47. https://doi.org/10.1109/MWC.004.2300572.