Scholar Publishing Group
International Journal of Multimedia Computing
https://doi.org/10.38007/IJMC.2025.060110
ISSN 2789-7168 Vol. 6, Issue 1:102-110

Architecture Design and Performance Optimization of Vehicle Intelligent Analysis System Based on Deep Learning and Edge Cloud Collaboration

Linghong Cheng

Security Org, Microsoft,Redmond, 98052, WA, US

Keywords: Cloud edge collaboration; Task uninstallation; Image recognition; Federated Learning; Sample imbalance.

Abstract: The integration of cloud computing, big data, artificial intelligence, and image recognition technologies is driving the intelligent upgrading of various industries. Cloud edge collaborative computing has become a key supporting technology for artificial intelligence image recognition by integrating the strong computing/storage capabilities of the cloud with the low latency real-time processing advantages of the edge. However, existing research has significant shortcomings in the integrated process of model training inference, joint inference strategies for edge resource constrained scenarios, and imbalanced samples in federated learning, which restrict its efficient application. To this end, this article focuses on three aspects of research in the field of artificial intelligence image recognition: firstly, a cloud edge collaborative image training and inference integrated task offloading model based on Kubernetes/Kubeedge framework is constructed, which realizes the full process automation deployment of cloud model training, mirror issuance, and edge inference; Secondly, a resource constrained cloud edge collaborative inference task offloading strategy is proposed, which triggers collaborative inference by monitoring edge load overruns and inference probability values; Finally, to address the issue of imbalanced samples in federated learning, a comprehensive weight evaluation method based on local model accuracy, stability, and sample size is proposed to optimize global model aggregation. The experimental results show that the integrated process can reduce data transmission delay and improve inference response speed. The joint inference strategy has significantly better inference efficiency and accuracy than traditional methods in medical pathology and marine fish image classification scenarios. The federated learning aggregation method effectively weakens the influence of sample differences and improves the accuracy of the global model in imbalanced sample scenarios. This research provides a reusable cloud side collaborative architecture design and performance optimization scheme for the vehicle intelligent analysis system, balancing low latency, high accuracy and privacy protection requirements. In the future, it will expand the task resource data collaboration mechanism of the Internet of Things scene, explore the reasoning mechanism of the complex scene of the industrial Internet, and optimize the accuracy of the global model by combining sample imbalance processing and aggregation methods, to promote the efficient application of cloud side collaborative computing in a wider range of scenarios.

1 Introduction

The integration of cloud computing, big data, artificial intelligence, and image recognition technology has promoted the intelligent upgrading of various industries, especially in fields such as healthcare, ocean, and smart cities. Cloud edge collaborative computing has become a key supporting technology for artificial intelligence image recognition by integrating the strong computing/storage capabilities of the cloud and the low latency real-time processing advantages of the edge. However, existing research still has significant shortcomings in the integrated process of model training inference, joint inference strategies in edge resource constrained scenarios, and sample imbalance in federated learning. Model separation leads to inefficient processes, joint inference relies heavily on model segmentation or single accuracy threshold decisions, data non independent and identically distributed in federated learning and sample imbalance lead to local model contribution bias, and traditional data augmentation or sharing methods are prone to privacy violations. To address these challenges, this article focuses on the architecture design and performance optimization of a vehicle intelligent analysis system based on deep learning and edge cloud collaboration. By constructing an integrated process of cloud model training based on Kubernetes/Kubeedge framework, mirroring and issuing to the edge for inference, the difficulty of model online updates is solved; Design a dynamic task offloading strategy that combines load overload and inference probability threshold to optimize joint inference performance in edge resource constrained scenarios; Propose a comprehensive weight evaluation method based on local model accuracy, stability, and sample size, improve federated learning model aggregation, and enhance the accuracy of the global model in sample imbalance scenarios. The theoretical innovation of this article lies in formally describing the correlation between cloud edge collaborative tasks, resources, and data, and improving the integrated process model and joint reasoning strategy; The technological breakthrough is reflected in the development of plugins based on the Kubernetes/Kubeedge/Sedna framework and their integration into the cloud edge collaboration platform. Through multi scenario verification such as medical pathology images and marine fish images, the performance is superior to traditional solutions; The practical value lies in providing reusable cloud edge collaborative architecture design and performance optimization solutions for vehicle intelligent analysis systems, balancing low latency, high precision, and privacy protection requirements, and promoting the implementation and application of low latency, high precision, safe and reliable intelligent vehicle analysis systems.

2 Correlation theory

2.1 Analysis of Cloud Edge Collaboration Technology Fusion Architecture

Cloud computing[1]~[7], with its ultra large scale, virtualization, high scalability, and containerization orchestration capabilities, provides dynamic and scalable computing resources and storage support for enterprises, becoming the core infrastructure for massive data processing; Edge computing significantly reduces transmission delay[10]~[12] and improves response speed through localized processing of data sources. It is suitable for Internet of Things [10]~[12] and mobile data scenarios. Although it faces resource constrained challenges, its processing capacity continues to increase with the coordinated development of cloud edge. Cloud edge collaborative computing integrates the advantages of both, optimizing system performance through task offloading and resource allocation - processing complex tasks in the cloud, achieving fast response at the edge, while enhancing privacy protection through localized data processing, reducing energy consumption, and improving operational efficiency. This technology has been widely applied in scenarios such as image recognition model training, edge inference, joint inference, and federated learning, forming a collaborative system of "cloud strong computing edge low latency"[10]~[12] , becoming a key technical support for promoting intelligent upgrading in various industries, especially in intelligent analysis scenarios that require low latency and high privacy protection.

2.2Analysis of Cloud Edge Collaborative Task Unloading and Federated Learning Model Aggregation Technology

Cloud based training edge inference task offloading achieves integrated operations of model training, image generation and issuance, and inference tasks through cloud edge collaborative architecture. Utilizing YAML files [10]~[12] to schedule cloud resources such as CPU, memory, Pod container quantity, and data paths, task execution efficiency and resource utilization are optimized, and data transmission latency is reduced; Edge inference relies on edge nodes to perform offloading tasks, reducing cloud burden and shortening response time. To address the issue of limited edge node resources, cloud edge collaborative joint inference tasks are offloaded using model segmentation methods (shallow network edge processing is transmitted to cloud based deep inference) or dynamic model selection algorithms (lightweight models are deployed at the edge and whether to upload to the cloud is determined based on inference probability thresholds), but the problem of frequent interactions or incomplete judgment conditions needs to be addressed. Cloud edge collaborative federation learning model aggregation generates a global model by aggregating local models locally trained by each participant through the central server, which is divided into horizontal federation (with the same characteristics and different labels) and vertical federation (with the same labels and different characteristics). The traditional FedAvg algorithm[10]~[12] is weighted based on the sample proportion. When the samples are unbalanced or the quality of local models is different, it needs to combine model evaluation and high-quality model mining to adjust the weight to eliminate the impact of data islands and sample deviations. Kubernetes, as an open-source container orchestration platform, implements cluster interface management, resource scheduling, storage configuration, and load balancing through components such as API Server[10] , Scheduler, ETCD, Kube proxy, etc. It can be combined with Kubeedge to build a cloud edge collaborative environment; Kubeedge extends edge computing capabilities based on Kubernetes, supports container application deployment to the edge layer, provides node authentication, task unloading, load balancing and cloud edge data synchronization functions, and forms a "cloud edge end" integrated architecture; As a sub project of Kubeedge, Sedna supports modular development and is compatible with mainstream AI frameworks such as Tensorflow and Pytorch. It interfaces with Kubernetes and Kubeedge through interfaces such as GlobalManager API and LocalController API to achieve rapid updates and deployment of functional modules. The three together form the technological foundation of cloud edge collaborative computing, promoting the implementation of low latency and high scalability intelligent applications.

3 Research method

3.1 Research on Integrity and Security Protection of Cloud Storage Data

Cloud edge collaborative computing integrates the strong computing power of the cloud with the low latency characteristics of the edge, achieving a collaborative mode of cloud model training and edge inference, effectively alleviating the network bandwidth pressure and response delay caused by cloud inference, especially suitable for delay sensitive scenarios such as medical images. However, the traditional process of separating training and inference has problems such as cumbersome transformation and deployment, frequent cloud edge interactions, and low execution efficiency. It is necessary to build an integrated task offloading model to simplify the process and optimize resource allocation. This article proposes an image training and inference integrated task offloading model based on cloud edge architecture, as shown in Figure 1,

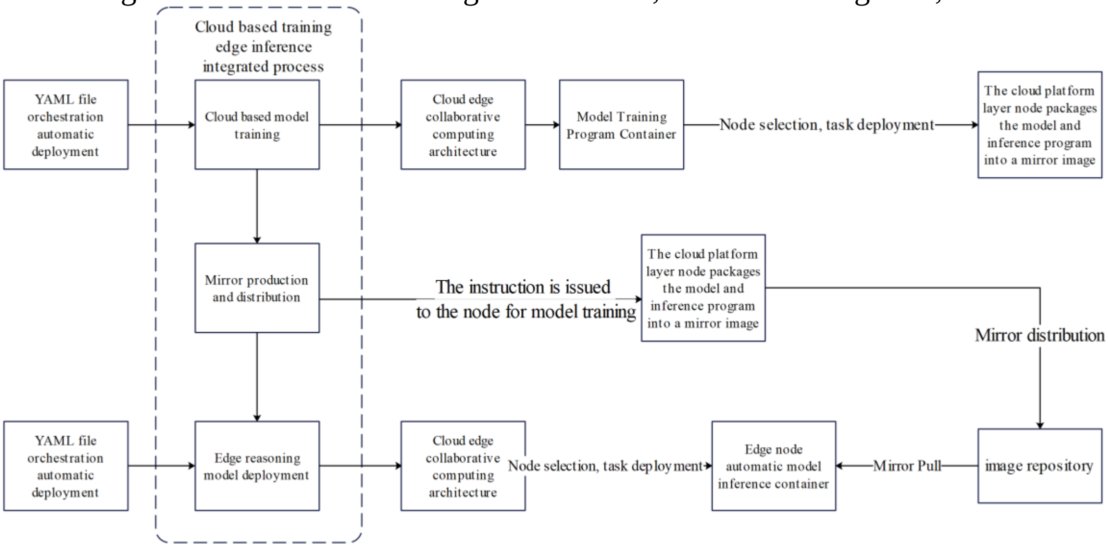

Figure 1 Task offloading model integrating cloud edge architecture image training and inference

The model consists of three core steps: cloud model training, model image creation and distribution, and edge inference. Through YAML orchestration and automatic deployment technology, a training container is created in the cloud based on deep learning frameworks and datasets. CPU and memory resource requirements are specified, and after completing model training, executable programs are packaged using Pyinstaller. Container images are generated using Docker and stored in the image repository; Edge nodes apply for resources through YAML files, pull the image, and start the inference container to perform image recognition. In terms of formal modeling, define a set of cloud edge nodes

$$w = \{w_1, \cdots, w_n\}$$

Node available resources $s_w = \left[s_{w_1}^{dpv}, s_{w_1}^{nsz}, \cdots, s_{w_n}^{dpv}, s_{w_N}^{nsz},\right]^T$; The set of homework tasks j_k includes o

Each assignment contains c tasks, which carry data and resource requirements $V_K = \left\{V_K^{dqv}, V_K^{nsz}\right\}$

after the task is submitted, the master node determines the node resources s_w and task requirements V_K based on thew'match uninstallable nodes and optimize the objective function

$$\mathrm{nio}(I_{w'1} b_{w'} + I_{w'2} C_{w'}) E_k$$

Minimize task execution time and data transmission delay to achieve efficient offloading under task resource collaborative allocation. This model automates the training inference process through containerization, YAML orchestration, and other technologies, ensuring the efficiency of model development and deployment, as well as the accuracy of inference speed. It provides theoretical support and technical path for intelligent applications in cloud edge collaboration scenarios.

3.2 Research on Integrity Verification and Fast Recovery Mechanism for Multi Cloud Data Migration

The integrated task offloading of image training and inference based on cloud edge architecture achieves task resource data collaborative allocation through YAML files that integrate resource application and Pod container offloading, in order to minimize task execution time and data transmission delay. In the training stage of the image recognition model, a CNN-LSTM neural network model is used. After normalizing the pathological images as input, the convolutional layer extracts features through convolutional kernels

$$x_j^{(l)} = \int \left(\sum_{i \in M^{i-1}} x_i^{(l-1)} * k_{ij}^{(l)} + b_j^{(L)} \right)$$

After dimensionality reduction in the pooling layer, it is passed to the LSTM layer, and the parameters are corrected by backpropagation using the Adam algorithm. Finally, training is completed through Softmax classification. After the training is completed, Pyinstaller packages the code to generate an executable program, and Docker builds an image medical track: latest based on Dockerfile and stores it in the image repository. During the image distribution phase, the medical test image is generated by importing the inference program and dependency environment through Dockerfile. When unloading inference tasks, YAML files specify deployment to the edge layer EdgeNode node, control the number of Pods through minReplicas/maxReplicas, apply for 1 core CPU/1GB memory resources with the Resources Requests tag, pull image files with the Image tag, and configure data reading and result output paths with the Env tag. After the task is submitted, the master node matches the resource requirements (such as training tasks requiring 4 cores of CPU/4GB of memory, and inference tasks requiring 1 core of CPU/1GB of memory), selects the node with the smallest delay execution time from the available node queue to complete the uninstallation, and achieves efficient collaboration between cloud training and edge inference.

3.3 Analysis of Experimental Results of Image Training and Reasoning Integration Based on Cloud Edge Architecture

The experimental environment is built on a supercomputing platform, including 1 master node (8CPU/16GB), 5 node nodes (8CPU/8GB), 1 file storage, 1 image repository, and 3 EdgeNode edge nodes (2CPU/2GB to 4CPU/4GB). The cloud edge collaboration performance is verified using 20000 medical pathology images (16000 training sets/4000 test sets). Experiment 1: Implementation of cloud training edge inference integration process: Design an 8-layer CNN-LSTM model (3-layer convolutional layer/2-layer pooling layer/1-layer LSTM layer/fully connected layer/output layer) with 5×5 convolutional kernels, 1 step size, and 3 numbers. After 20 epochs of training, the model accuracy reaches 0.7849; After the training is completed, a medical transfer: latest image is generated using Dockerfile and stored in the image repository. The edge node pulls the image and starts the Pod to complete the inference. Experiment 2: Comparison of cloud and edge inference performance: In 20 experiments, the total time for edge inference was less than that in the cloud, due to the low latency of edge data transmission (VPN simulation bandwidth of 20M, significantly shorter transmission time than the 10M bandwidth of public IP); The pure inference time is similar, but the difference in data transmission time is significant (edge transmission time 33s-27s vs cloud 84s-123s). Specific cases show that the total edge time of 284 seconds (inference 241s+transmission 33s) in the first experiment was 12.9% faster than 326 seconds in the cloud, and the total edge time of 260 seconds in the 20th experiment was 25% faster than 348 seconds in the cloud, mainly due to low latency transmission at the edge. Experiments have shown that the cloud edge collaborative integration process can efficiently deploy models, and edge inference significantly improves task execution efficiency and response speed by reducing long-distance transmission delays, verifying the effectiveness of cloud edge architecture in time sensitive scenarios such as medical pathology image recognition.

4 Results and discussion

4.1Improving the System Model of CP-ABE Data Sharing Scheme

For resource constrained scenarios, cloud edge collaborative image inference task offloading optimizes task allocation through a dual trigger strategy of load overload and low recognition probability. When the load exceeds the limit, calculate the CPU/memory resource limit parameter through the formula G_1: if $T_w^{dqv} \leq 0$ or $T_w^{nsz} \leq 0$, $G_1 = 1$ pause the task and check the network transmission status; Calculate the minimum transmission time U_{nj} and actual transmission time U_{sfbm} according to the formula, if $U_{sfbm} \leq U_{nj}/2$.Then, trigger cloud migration. Extracting the maximum probability value of an image after edge inference in low probability scenarios$B$ If$B < B'$ so$G_3 = 1$, trigger secondary cloud inference. The algorithm is implemented based on the Sedna plugin . By configuring the apiVersion to Sedna. io/v1 alpha 1 using YAML, the Sedna interface is called and the joint_inference: latest image is deployed. Global variables such as load threshold and network bandwidth are integrated to achieve resource network accuracy collaborative optimization. Experimental verification shows that this method can effectively reduce edge resource pressure, improve inference efficiency and recognition accuracy, and is suitable for large image and high real-time demand scenarios such as medical pathology and marine biology.

4.2 Model experiment

The experimental design of a resource constrained cloud edge collaborative image inference task offloading method involves deploying Kubernetes Kubeedge Sedna architecture on a supercomputing platform, where the main node (8CPU/16GB memory) and EdgeNode (2CPU/2GB memory) are connected through a 2M bandwidth to simulate high latency transmission. Two scenarios were tested: (1) Medical pathology image recognition, using 4000 esophageal cancer images, with 2000 images randomly selected from each group, for a total of 10 groups. EdgeNode deployed a lightweight model (batch size of 20, up to 3 concurrent tasks), while the main node deployed a high-precision model. (2) Ocean fish image recognition, using 2000 images, randomly select 10 groups of 1000 images per group according to a similar deployment. Both scenarios compared the proposed cloud edge collaborative inference algorithm with Sedna's default joint inference algorithm, recording the total completion time, EdgeNode CPU/memory utilization, data transfer volume, and recognition accuracy. The results indicate that in both cases, the proposed algorithm outperforms Sedna's default algorithm. For medical pathology (Group 3), the proposed method reduced total time by 21% (442 seconds vs. 559 seconds), EdgeNode processing time by 23.4% (394 seconds vs. 515 seconds), CPU/memory utilization by 11.3%/14.6% (70%/76% vs. 79%/89%), while improving accuracy by 3.6% (69.7% vs. 66.1%), and offloading 257 images to the cloud. For marine fish (Group 5), it reduced the total time by 26.8% (486 seconds vs. 664 seconds), EdgeNode processing time by 24.6% (456 seconds vs. 605 seconds), CPU/memory utilization by 12.1%/12.9% (72%/81% vs. 82%/93%), unloaded 94 images, and achieved higher accuracy. These improvements stem from the algorithm's dual strategy: (1) offloading tasks when EdgeNode resources exceed a threshold (70% CPU/75% memory) to avoid overload and maintain efficient processing; (2) Unload low confidence images (below tolerance threshold, such as medical 0.55/ocean 0.5) to a high-precision model in the cloud for re identification to improve accuracy. This method effectively balances the resource utilization rate and task efficiency of the EdgeNode, and alleviates the problem of slow reasoning speed and low precision caused by the limited edge computing capability. By dynamically adjusting task offloading and utilizing cloud edge resources for collaboration, it improves inference efficiency and accuracy, and enhances the service quality of edge intelligent applications.

4.3 Effect analysis

The experiment is based on the deployment of Kubernetes Kubeedge Sedna cloud edge collaborative architecture, including one master node (8CPU/16GB memory) and three participant nodes (each 4CPU/4GB memory+Tesla T4 GPU). The esophageal early cancer pathology image dataset (20000 images, approximately 6500 normal/low/high-level samples each) is used, and 4000 images are extracted as the common test set. The remaining 16000 images are shown in Table 1

Table1 Sample Data Allocation Details

Experiment Group	Participant-1 (Normal, Low, High)	Participant-2 (Normal, Low, High)	Participant-3 (Normal, Low, High)
Group 1	3400, 200, 200	200, 3400, 200	1000, 1000, 2000
Group 2	550, 1000, 400	500, 300, 2000	250, 500, 2200
Group 3	1750, 1750, 1250	1650, 1850, 1250	1550, 1550, 1650
Group 4	1650, 1700, 1670	1650, 1720, 1650	1620, 1715, 1685
Group 5	150, 1800, 1800	100, 1800, 1900	150, 1500, 1300
Group 6	1000, 1000, 1000	1500, 1500, 1500	2000, 2000, 2000
Group 7	100, 3400, 1400	1200, 1000, 1300	3600, 900, 2900
Group 8	1200, 1500, 2200	1900, 2100, 1500	2100, 1700, 1700
Group 9	3000, 200, 200	1000, 1200, 1400	1000, 1300, 1300
Group 10	200, 1500, 500	500, 1500, 1000	200, 1000, 2500

Randomly allocate to 3 participants to simulate an imbalanced sample scenario (e.g. Group 1: Participant 1 holds 3400 normal level+200 low/high level; Participant 2 holds 200 normal level+3400 low level+200 high level). Ten experimental groups were set up to compare the global model accuracy of the cloud edge collaborative model aggregation method in this chapter with Sedna's default FedAvg method. The average global model accuracy of the method in this chapter was higher than that of FedAvg. Specifically, the accuracy of the first group experiment was 52.7% (FedAvg 49.2%), due to the fact that Participant 3 had a local model stability weight of 57% and a contribution value weight of 41% (Table 2)

Table 2 Weight Allocation of Local Models in Cloud-Edge Coordinated Aggregation

Participant	Max Accuracy Weight	Stability Weight	Sample Proportion Weight	Contribution Weight
Participant-1	36%	21%	33%	30%
Participant-2	34%	22%	33%	29%
Participant-3	30%	57%	34%	41%

Lead global aggregation; The accuracy of the sixth experiment was 58.2% (FedAvg 55.1%), with a sample size of 2000 participants and a maximum recognition accuracy weight of 42%+stability weight of 63% (as shown in Table 3)

Table3 6th Exp. Cloud-Edge Aggregation Local Model Weight

Participant	Max Accuracy Weight	Stability Weight	Sample Proportion Weight	Contribution Weight
Participant-1	27%	20%	23%	23%
Participant-2	31%	17%	33%	27%

Participant-3	42%	63%	44%	50%

Contribution value weight of 50%, reducing the impact of low-quality models; The accuracy of the third experiment was 57.7% (FedAvg 56.2%), the stability weight of Participant 3 was 62%, the contribution weight was 49%, and the overall recognition was stable. The analysis shows that the method proposed in this chapter effectively alleviates the global model bias caused by sample imbalance through local model quality evaluation and contribution value weight adjustment. Experimental data validates the effectiveness of this method in complex sample scenarios such as medical pathology.

5 Conclusion

This article focuses on the research of cloud edge collaborative task offloading methods in the context of artificial intelligence image recognition. Innovative achievements have been made in the integrated process of model training inference, optimization of inference in resource limited scenarios, and the problem of imbalanced federated learning samples. The study first constructed an image training and inference integrated task offloading model based on cloud edge architecture, and implemented the full process automation deployment of cloud model training, mirror issuance, and edge inference through Kubernetes/Kubeedge framework. Experimental verification showed that it can reduce data transmission latency and improve inference response speed; Secondly, a resource constrained cloud edge collaborative inference task offloading strategy is proposed, which triggers collaborative inference by monitoring edge load overload and inference probability values. After integration into the Sedna platform, the inference efficiency and accuracy are significantly better than traditional methods in medical pathology and marine fish image classification scenarios; Finally, to address the issue of imbalanced samples in federated learning, a method for evaluating aggregated weights that integrates local model accuracy, stability, and sample size is proposed. This method has been proven effective in reducing the impact of sample differences on the global model and improving model accuracy in multi scenario experiments on medical pathology datasets. Future research will focus on three major directions: firstly, expanding the collaborative mechanism of task resource data in the context of the Internet of Things to further improve execution efficiency; The second is to explore the cloud side collaborative reasoning mechanism in the complex scene of industrial Internet; The third is to combine sample imbalance processing and aggregation methods to optimize the accuracy of the global model, and promote the efficient application of cloud edge collaborative computing in a wider range of scenarios.

References

[1] Saillenfest A , Lemberger P .Nonlinear Concept Erasure: a Density Matching Approach[J]. 2025.

[2] Cui, N. (2025). The Practical Application of Traffic Flow Forecasting and Capacity Analysis. Journal of Computer, Signal, and System Research, 2(5), 65-71.

[3] Wang Z J .Closed-form solution-based fuzzy utility vectors acquired from trapezoidal fuzzy pairwise comparison matrices using logarithmic quadratic programming for improving fuzzy AHP decision-making systems[J].Journal of Computational and Applied Mathematics, 2025, 468.DOI:10.1016/j.cam.2025.116647.

[4] Hao, L. (2025). Research on Perception and Control System of Small Autonomous Driving Vehicles. International Journal of Engineering Advances, 2(2), 48-54.

[5] Wu X, Bao W. Research on the Design of a Blockchain Logistics Information Platform Based on Reputation Proof Consensus Algorithm[J]. Procedia Computer Science, 2025, 262: 973-981.

[6] Pan, H. (2025). Development and Optimization of Social Network Systems on Machine Learning. European Journal of AI, Computing & Informatics, 1(2), 73-79.

[7] Yu B , Yuan H , Li H ,et al.Long-Short Chain-of-Thought Mixture Supervised Fine-Tuning Eliciting Efficient Reasoning in Large Language Models[J]. 2025.

[8] Yang D, Liu X. Collaborative Algorithm for User Trust and Data Security Based on Blockchain and Machine Learning[J]. Procedia Computer Science, 2025, 262: 757-765.

[9] Cai, Y. (2025). Research on Positioning Technology of Smart Home Devices Based on Internet of Things. European Journal of AI, Computing & Informatics, 1(2), 80-86.

[10] Zillmann P , Garavaglia A , Schenk A M ,et al.Reducing channel interference with complex-valued asymmetrical weighted overlap and add filtering:US17805493;US202200017805493;US202217805493A;US202217805493[P].US12255851B2;US2025012255851B2;US12255851B2;US12255851[2025-08-20].

[11] Lu, C. (2025). The Application of Point Cloud Data Registration Algorithm Optimization in Smart City Infrastructure. European Journal of Engineering and Technologies, 1(1), 39-45.

[12] Madhavi S , Praveen R , Jagatheswari S ,et al.Hybrid ELECTRE and bipolar fuzzy PROMOTHEE-based packet dropping malicious node mitigation technique for improving QoS in WSNs[J].International Journal of Communication Systems, 2025, 38(2).DOI:10.1002/dac.5974.

[13] Jing, X. (2025). Research on the Application of Machine Learning in the Pricing of Cash Deposit Products. European Journal of Business, Economics & Management, 1(2), 150-157.

[14] Tang X, Wu X, Bao W. Intelligent Prediction-Inventory-Scheduling Closed-Loop Nearshore Supply Chain Decision System[J]. Advances in Management and Intelligent Technologies, 2025, 1(4).

[15] Zhu, Z. (2025). Application of Database Performance Optimization Technology in Large-Scale AI Infrastructure. European Journal of Engineering and Technologies, 1(1), 60-67.