ISSN 3107-7657 (Print) ISSN 3107-7657 (Online)

Research Article

Optimizing Large-Scale AI Server Testing via Adaptive Evolutionary Algorithms

Xingcheng Ren[1,*]

Quanta Manufacturing Nashville LLC, TN 37086, USA

* Corresponding author:Xingcheng Ren

xrenwork@yahoo.com

Abstract: Traditional testing methods are inefficient in high-dimensional parameter space, dynamic load fluctuation and heterogeneous hardware architecture, and it is difficult to meet the testing requirements of large-scale clusters. In this paper, an Environment-Aware Adaptive Hybrid Algorithm is proposed, which can effectively solve the convergence stagnation problem of high-dimensional parameter space through dynamic mutation rate adjustment and hybrid coding strategy. EA-AHA adopts a master-slave island model, in which the master island is responsible for global exploration and the slave island is responsible for local exploitation, and maintains population diversity through individual migration. The algorithm uses chromosome representation coded by real numbers and integers, and uses adaptive mutation rate and crossover strategy to adapt to different test scenarios. The multi-objective fitness function comprehensively considers task execution time, system resource utilization rate and fault detection rate, so as to realize the synergistic improvement of test efficiency, resource utilization rate and fault detection rate. In addition, EA-AHA models the optimization problem as a dynamic optimization problem, and responds to dynamic changes such as background load fluctuation and hardware performance attenuation in the test environment through environmental state awareness and adaptive response mechanism. The closed-loop mechanism of simulation and physical verification further improves the optimization efficiency and practical effectiveness. The experimental results show that EA-AHA is superior to Bayesian optimization and NSGA-II algorithm in convergence speed, comprehensive performance and dynamic environment adaptability, which effectively solves the problem of large-scale AI server test optimization.

Keywords: *Adaptive evolutionary algorithm; AI server test; EA-AHA;*

1. Introduction

With the parameter scale of the generative AI model exceeding one trillion, the super-large-scale server cluster supporting its training faces severe testing challenges. This surge in model complexity necessitates hyperscale data center architectures capable of meeting unprecedented computational demands [4]. Traditional testing methods have exposed bottlenecks such as combination explosion, environmental mismatch and low optimization coverage when dealing with high-dimensional adjustable parameters, dynamic load fluctuation and heterogeneous hardware architecture. The efficiency of existing optimization technologies such as Bayesian optimization and NSGA-II is significantly reduced in high-dimensional or dynamic scenarios, and the expert system lacks the ability of cross-platform migration, which makes it difficult to meet the requirements of efficient and stable testing of thousand-node clusters. This study focuses on the optimization of AI server cluster test in ultra-high dimensional parameter space, and realizes the synergistic improvement of test efficiency, resource utilization and fault detection rate. An adaptive evolutionary framework based on environmental awareness is proposed, which solves the convergence stagnation problem of high-dimensional parameter space through dynamic mutation rate adjustment and mixed coding strategy. Such frameworks are essential for constructing automated testing processes that can scale with the increasing complexity of AI infrastructure [1]. Notably, the advancement of such high-dimensional optimization frameworks is significantly propelled by government tax preferences, which act as a vital catalyst for independent innovation within emerging technology enterprises [5].

2. Design of adaptive evolutionary algorithm

ISSN 3107-7657 (Print) ISSN 3107-7657 (Online)

2.1 Algorithm core framework

An Environment-Aware Adaptive Hybrid Algorithm (EA-AHA) is studied and designed. Based on the framework of the classical differential evolution (DE) algorithm, this algorithm introduces the environment-aware feedback mechanism, hybrid coding strategy and adaptive parameter adjustment to effectively deal with the test optimization problem of AI server.

As shown in Figure 1, EA-AHA adopts the master-slave island model as the basic framework, in which one master island is responsible for global exploration and multiple slave islands (subpopulations) are responsible for local exploration for different test scenarios. Individuals migrate regularly between islands to avoid premature convergence and maintain population diversity.

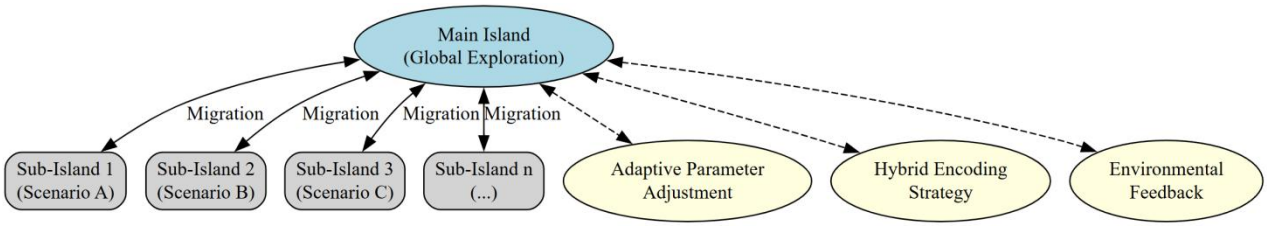

Figure 1. Schematic diagram of EA-AHA algorithm framework

2.2 Hybrid coding strategy

In order to solve the mixed optimization problem of heterogeneous hardware (CPU, GPU, NVLink, InfiniBand) and software parameters, such as batch size, learning rate and MPI thread number, a chromosome representation with mixed encoding of real numbers and integers is adopted. Optimizing performance across such distributed and heterogeneous systems requires addressing unique bottlenecks in parallel computing environments [3] [6].An individual (chromosome) is represented as a mixed coding vector X, which represents a complete set of server cluster test configuration parameters. $X=[x_1,x_2,\cdots,x_d,x_{d+1},\cdots,x_n]$, Where $x_1,x_2,\cdots,x_d$ is the real number coding parameter. $x_{d+1},\cdots,x_n$ is an integer coding parameter, such as Batch Size, MPI process number, GPU memory allocation strategy (enumeration type), network topology binding strategy, etc.

Adaptive mutation and crossover

The mutation rate F is no longer a fixed value, but is dynamically adjusted according to the optimization progress of the previous generation. This approach aligns with adaptive supervised learning techniques designed for handling data sparsity and dynamic constraints in high-dimensional spaces [23].

$$F_{g+1}=F_{\min}+(F_{\max}-F_{\min})e^{-\lambda(\Delta f_g/f_g)} \quad (1)$$

Among them, F_{g+1} is the variation rate used by the next generation population. $F_{max},F_{min}$is the preset upper and lower bounds of the variation rate. λ is the attenuation coefficient, which controls the sensitivity of adjustment, and is usually set to 1. Δf_g is the variation of the optimal fitness in the g generation population. f_g is the average fitness of the g generation population.

When the optimization progress is significant (Δf_g is large), the exponential term approaches 0, F approaches F_{min}, and the algorithm tends to local fine search. When stuck in stagnation (Δf_g is small or negative), the exponential term approaches 1, and F approaches F_{max}. The algorithm increases the mutation rate to jump out of the local optimum and enhance the global exploration ability.

For the real coding part, the binomial crossover of DE algorithm is adopted. For the integer coding part, uniform crossover integer variants are used to ensure the generation of effective integer candidate solutions.

ISSN 3107-7657 (Print) ISSN 3107-7657 (Online)

Multi-objective fitness function

The essence of test optimization is a multi-objective problem, and the fitness function $F(X)$ is designed as the weighted harmonic average of three key indicators to balance the needs of different test objectives.

$$F(X)=\alpha\frac{1}{T_{exec}}+\beta R_{util}+\gamma F_{detect} \quad (2)$$

Where T_{exec} represents the task execution time. The total time required to run the standard benchmark program under configuration X. The goal is to minimize it, so take its reciprocal. R_{util} stands for system resource utilization. The weighted average of cluster average CPU, GPU and network bandwidth utilization during the test period. The goal is to maximize. F_{detect} stands for failure detection rate. The number of potential hardware/system failures triggered or detected under this configuration (to be normalized). The goal is to maximize. α,β,γ stands for weight coefficient. It can be dynamically adjusted according to the specific objectives of the testing stage. For example, γ can be increased in the stability testing phase and α,β can be increased in the performance testing phase.

3. Large-scale AI server test scenario modeling

3.1 Test parameter space modeling

Large-scale AI server testing is defined as a high-dimensional, mixed and constrained optimization problem. That is, the algorithm individual X, its dimension n may be as high as tens or even hundreds of dimensions, including all adjustable system BIOS parameters, OS kernel parameters, deep learning framework parameters, distributed training library parameters and so on. Each parameter has its feasible region, which is defined as $x_i\in[L_i,U_i]$, where L_i,U_i is the lower and upper bounds of the technical permission of parameter x_i, respectively.

In the optimization process, the constraints are integrated into the fitness evaluation by the penalty function method to ensure the feasibility of the solution; Where the equality or inequality constraint is expressed as $g_j(X)\le 0$, when the constraint is violated, a quadratic penalty term is introduced:

$$Penalty(X)=\rho\sum_j\left(\max(0,g_j(X))\right)^2 \quad (3)$$

And deducted from the original fitness to form a modified fitness function:

$$F'(X)=F(X)-Penalty(X) \quad (4)$$

Among them, the punishment factor ρ controls the punishment intensity of infeasible solutions, thus guiding the search to converge to the feasible region that meets the resource constraints.

Dynamic environment modeling

In order to cope with dynamic changes such as background load fluctuation and hardware performance attenuation in the test environment, this paper models the optimization problem as a dynamic optimization problem (DOP), and realizes continuous optimization through environmental state awareness and adaptive response mechanism. Implementing fault-tolerant and real-time

ISSN 3107-7657 (Print) ISSN 3107-7657 (Online)

scheduling is particularly critical for maintaining stability in critical AI infrastructure under such fluctuations [14][27].The algorithm collects the environment state vector S_t including CPU, memory load, GPU temperature and other information before each generation of evolution, and triggers the response strategy immediately when the significant change of $S_t$is detected. Firstly, the fitness of the current population is re-evaluated to reflect the real performance in the new environment. At the same time, the diversity restoration mechanism is introduced to retain elite individuals and inject random new individuals to enhance the population's ability to explore the new environment, thus ensuring that the optimization process is still efficient and robust under dynamic conditions.

Closed loop of simulation and physical verification

Firstly, EA-AHA algorithm is run in the digital twin simulation layer based on proxy model for low-cost and coarse-grained search, and the model trained by historical data is used to predict the configuration performance and quickly screen out the potential elite parameter subset. Then, these candidates are configured on real clusters for high-fidelity verification, and accurate performance and resource indicators are obtained, and the actual results and newly discovered failure modes are fed back to the simulation layer for dynamic calibration and optimization of the proxy model, thus forming a closed-loop iterative mechanism of "simulation optimization-physical verification-model update", giving consideration to optimization efficiency and actual effectiveness. Similar data-driven decision-making models have been successfully utilized to optimize resource allocation and marketing budgets in other complex industrial contexts [20][21]. To ensure the long-term viability of this iterative cycle, it is essential to implement an effective teamwork incentive mechanism—specifically one that accounts for participants' unfairness aversion—to optimize the collaborative performance between hardware engineers and algorithm designers [8].

4. Experimental results and analysis

In order to verify the effectiveness of EA-AHA algorithm, this study conducted experiments on an AI computing cluster with 256 nodes. The cluster is equipped with NVIDIA A100 GPU and InfiniBand HDR high-speed network. EA-AHA is compared with two mainstream optimization algorithms-Bayesian optimization (BO) and standard NSGA-II algorithm.

The experiment is first carried out in a digital twin simulation environment to evaluate the convergence characteristics of each algorithm at low cost. The optimization goal is to minimize the task execution time (T_{exec}), and the maximum number of evaluations is set to 1000 times. As shown in Figure 2, EA-AHA algorithm shows the fastest convergence speed, and approaches the global optimal region after about 400 evaluations, which is due to its adaptive mechanism of environment awareness and effectively avoids the early convergence stagnation in high-dimensional space. BO algorithm shows obvious "dimension disaster" in high-dimensional problems, and the cost of constructing its proxy model based on Gaussian process rises sharply with the increase of dimension, and the search efficiency is the lowest. The convergence speed of NSGA-II is moderate, but its fixed parameter strategy is not flexible enough in complex environment, and the later optimization progress is slow.

ISSN 3107-7657 (Print) ISSN 3107-7657 (Online)

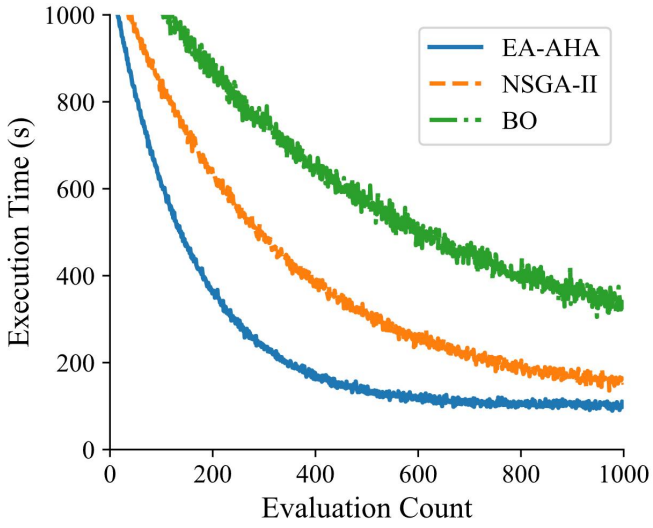

Figure 2. Comparison of convergence curves of various algorithms in simulation environment

Run each algorithm for 300 iterations in the physical measurement environment to comprehensively optimize the three objectives of T_{exec}, R_{util}, F_{detect} (the weight is set to $\alpha = 0.5, \beta = 0.3, \gamma = 0.2$). Table 1 below shows the performance data of the solution with the highest comprehensive score in Pareto frontier found by each algorithm.

Table 1. Comparison of optimal configuration performance of each algorithm in physical measurement environment

Algorithm	Task execution time (s) ↓	System resource utilization ratio (%) ↑	Fault detection rate (%) ↑	Comprehensive fitness ↑
EA-AHA	1123	92.5	5.8	0.842
NSGA-II	1256	89.1	3.2	0.781
BO	1386	85.7	1.5	0.723
Default configuration	1685	76.3	0.1	0.572

In the final performance, EA-AHA has achieved the best performance in all three optimization objectives, and its comprehensive fitness is significantly higher than that of the comparison algorithm. This proves the effectiveness of its hybrid coding strategy and multi-objective fitness function. The configuration found by EA-AHA successfully triggered multiple potential faults, and the fault detection rate (F_{detect}) far exceeds the comparison algorithm, which is very important for the stability test of large-scale clusters. BO's performance once again proves that it is not suitable for such high-dimensional and mixed variable optimization scenarios. The high detection rate is vital for ensuring the reliability of scalable storage and computing systems in AI-intensive applications [7] [9].

In order to test the adaptability of the algorithm to dynamic environment, two background load disturbances are artificially introduced in the optimization process. Figure 3 below shows the fitness fluctuation of EA-AHA and NSGA-II in dynamic environment. When the environment changes suddenly (gray area), the fitness of elite individuals of the two algorithms drops sharply. EA-AHA quickly triggered population reassessment and diversity injection through environmental awareness mechanism, and quickly recovered and found a new high-performance configuration within several generations, showing strong robustness. NSGA-II lacks an explicit environmental coping mechanism, and its recovery speed is slow and its performance fluctuates greatly.

ISSN 3107-7657 (Print) ISSN 3107-7657 (Online)

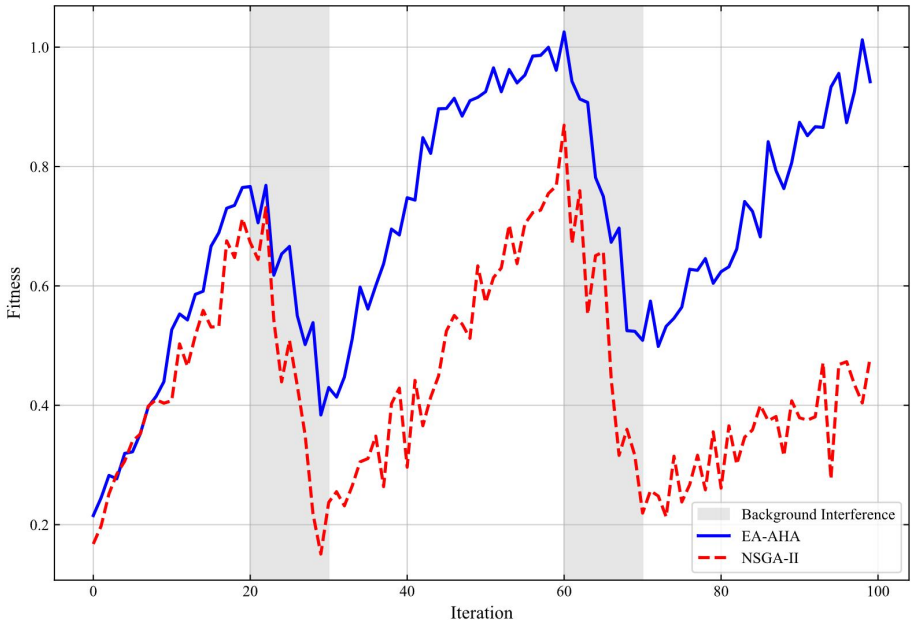

Figure 3. Comparison of performance stability of algorithms in dynamic environment

5. Conclusion

In this study, an EA-AHA algorithm is proposed to meet the challenge of ultra-large-scale server cluster testing caused by the parameter scale of generative AI model exceeding trillions. Based on the framework of classical DE algorithm, this algorithm introduces environment-aware feedback mechanism, hybrid coding strategy and adaptive parameter adjustment, which effectively addresses the problem of AI server test optimization. The experimental results show that EA-AHA algorithm has obvious advantages in both simulation environment and physical measurement environment. In the simulation environment, EA-AHA algorithm shows the fastest convergence speed, and can approach the global optimal region after about 400 evaluations, thus avoiding the early convergence stagnation in high-dimensional space. In the physical measurement environment, EA-AHA algorithm has achieved the best performance in the three objectives of task execution time, system resource utilization rate and fault detection rate, and its comprehensive fitness is significantly higher than that of the comparison algorithm. In addition, EA-AHA algorithm can quickly trigger population re-evaluation and diversity injection through environment awareness mechanism, which shows strong robustness in dynamic environment and can quickly recover and find new high-performance configurations. EA-AHA algorithm has significant application value in large-scale AI server test optimization, which can effectively improve test efficiency, resource utilization and fault detection rate, and provide strong support for efficient and stable testing of ultra-large-scale AI server clusters. Ultimately, the industrial adoption of EA-AHA must navigate the broader landscape of business cycles and financial shocks, recognizing that "animal spirits" and economic stability are pivotal factors in the widespread scaling of such computational innovations [2]. From a broader perspective, such technological advancements must also account for the volatility of business cycles and the incentive structures that drive independent innovation in emerging enterprises [2] [5].

Data Availability Statement

Data will be made available on request.

Funding

This work was supported without any funding.

Conflicts of Interest

The author(s) declare no conflicts of interest.

Ethical Approval and Consent to Participate

Not applicable.

ISSN 3107-7657 (Print) ISSN 3107-7657 (Online)

References

[1] Xingcheng, R. (2026). Research on the Construction and Application of Automated Framework for Large-scale AI Server Testing Process. International Journal of Computer Science and Engineering, 1(02), 55-61.

[2] Pang, F. (2025). Animal Spirit, Financial Shock and Business Cycle. European Journal of Business, Economics & Management, 1(2), 15-24.

[3] Perera, C. (2024). Optimizing performance in parallel and distributed computing systems for large-scale applications. Journal of Advanced Computing Systems, 4(9), 35-44.

[4] Sankar, T., Venkata Ramana, R. B., & Balamuralikrishnan, A. (2023). AI-Optimized Hyperscale Data Centers: Meeting the Rising Demands of Generative AI Workloads. International Journal of Trend in Scientific Research and Development, 7(1), 1504-1514.

[5] Pang, F. (2025). Research On The Incentive Effect Of Government Tax Preference On Independent Innovation Of Emerging Enterprises. European Journal of Business, Economics & Management, 27-34.

[6] Varma, Y., & Kothandaraman, M. (2022). Optimizing Large-Scale ML Training Using Cloud-Based Distributed Computing. International Journal of Artificial Intelligence, Data Science, and Machine Learning, 3(3), 45-54.

[7] Do, J., Ferreira, V. C., Bobarshad, H., Torabzadehkashi, M., Rezaei, S., Heydarigorji, A., ... & Alves, V. (2020). Cost-effective, energy-efficient, and scalable storage computing for large-scale AI applications. ACM Transactions on Storage (TOS), 16(4), 1-37.

[8] Pang, F. (2020, November). Research on Incentive Mechanism of Teamwork Based on Unfairness Aversion Preference Model. In 2020 2nd International Conference on Economic Management and Model Engineering (ICEMME) (pp. 944-948). IEEE.

[9] Priyadarshini, S., Sawant, T. N., Bhimrao Yadav, G., Premalatha, J., & Pawar, S. R. (2024). Enhancing security and scalability by AI/ML workload optimization in the cloud. Cluster Computing, 27(10), 13455-13469.

[10] Lin, A. (2026). Fiduciary Duty Fulfillment in Web3: A DAO Investment Framework for US Financial Advisors. International Academic Journal of Social Science, 2, 17-26.

[11] Wu, Y. (2026). A Study on the Impact of Cross-Departmental Data Collaboration on Marketing Campaign Efficiency in Fast-Moving Consumer Goods E-commerce: The Case of PepsiCo (China)'s 7UP and Mirinda Project. Frontiers in Management Science, 5(1), 7-12.

[12] Wang, P., Wang, H., Li, Q., Shen, D., & Liu, Y. (2024). Joint and individual component regression. Journal of Computational and Graphical Statistics, 33(3), 763-773.

[13] Gadde, H. (2022). AI in Dynamic Data Sharding for Optimized Performance in Large Databases. International Journal of Machine Learning Research in Cybersecurity and Artificial Intelligence, 13(1), 413-440.

[14] Hao, Z. (2025). Fault-Tolerant Real-Time Scheduling for Edge AI in US Critical Infrastructure. Engineering Frontiers, 1(4).

[15] Li, K., Chen, X., Song, T., Zhou, C., Liu, Z., Zhang, Z., ... & Shan, Q. (2025). Solving situation puzzles with large language model and external reformulation. arXiv preprint arXiv:2503.18394.

[16] Zhu, H., Luo, Y., Liu, Q., Fan, H., Song, T., Yu, C. W., & Du, B. (2019). Multistep flow prediction on car-sharing systems: A multi-graph convolutional neural network with attention mechanism. International Journal of Software Engineering and Knowledge Engineering, 29(11n12), 1727 – 1740.

[17] Wang, H., Li, Q., & Liu, Y. (2024). Multi-response Regression for Block-missing Multi-modal Data without Imputation. Statistica Sinica, 34(2), 527.

[18] Han, C. (2025). Can Language Models Follow Multiple Turns of Entangled Instructions?. arXiv preprint arXiv:2503.13222.

[19] Lin, A. (2025). Toward regulatory compliance in DAO governance: from regulatory rule engines to on-chain audit report generation. Journal of World Economy, 4(6), 12-20.

[20] Wang, C. (2026). A Study on Data-Driven Budget Optimization for US Enterprises' Cross-Border Marketing. Frontiers in Management Science, 5(1), 41-46.

[21] Wang, C. (2025). Research on the Precision Allocation of Cross-Border Marketing Resources of US Enterprises Driven by Digital Technology. Innovation in Science and Technology, 4(11), 7-13.

[22] Tao, Y., Jia, Y., Wang, N., & Wang, H. (2019, July). The fact: Taming latent factor models for explainability with factorization trees. In Proceedings of the 42nd international ACM SIGIR conference on research and development in information retrieval (pp. 295-304).

[23] Wang, H., Li, Q., & Liu, Y. (2023). Adaptive supervised learning on data streams in reproducing kernel Hilbert spaces with data sparsity constraint. Stat, 12(1), e514.

ISSN 3107-7657 (Print) ISSN 3107-7657 (Online)

[24] *Wang, H., Sun, W., & Liu, Y. (2022). Prioritizing autism risk genes using personalized graphical models estimated from single-cell rna-seq data. Journal of the American Statistical Association, 117(537), 38-51.*

[25] *Wu, Y. (2026). Research on the Impact of LinkedIn Business Account Data-Driven Operations on Brand Exposure of AI Startups—A Case Study of AristAI. International Academic Journal of Social Science, 2, 27-37.*

[26] *Lin, A. (2025). Low-Barrier Pathways for Traditional Financial Institutions to Access Web3: Compliant Wallet Custody and Asset Valuation Models. Frontiers in Management Science, 4(6), 80-86.*

[27] *Hao, Z. (2025). Task Affinity-Aware Scheduling for Multi-Core Edge Devices in Autonomous Vehicles. Engineering Frontiers, 1(2).*

[28] *Tao, Y., Wang, Z., Zhang, H., Wang, L., & Gu, J. (2025, July). Nevlp: Noise-robust framework for efficient vision-language pre-training. In International Conference on Intelligent Computing (pp. 74-85). Singapore: Springer Nature Singapore.*

[29] *Wu, Y. (2026). Research on Dynamic Prediction Model of Brand Marketing Content ROI Based on Machine Learning. International Journal of Advance in Applied Science Research, 5(2), 31-38.*

[30] *Hao, Z. (2026). Structure-Aware Deep Reinforcement Learning for Latency-Minimal Scheduling of Edge AI Inference on Heterogeneous Cores. Journal of Intelligence and Engineering Technology, 1(1), 50-59.*

[31] *Luo, M., Zhang, W., Song, T., Li, K., Zhu, H., Du, B., & Wen, H. (2021, January). Rebalancing expanding EV sharing systems with deep reinforcement learning. In Proceedings of the Twenty-Ninth International Conference on International Joint Conferences on Artificial Intelligence (pp. 1338-1344).*

[32] *Liu, Z., Jin, C., Li, S., Li, W., & Wang, J. (2024). Improvement for modeling the damping of the wake oscillator based on the Van der Pol scheme. Physics of Fluids, 36(7).*