2024 4th International Conference on Computer Science and Blockchain (CCSB)

# Research on Complex Control of Internet of Things Based on AI Agent Technology

Yulou Li *
CCCC, SECOND HARBOR ENGINEERING CO., LTD NO.5 BRANCH
Wuhan, Hubei
China
78431872@qq.com

Zhangzhi Tao
CCCC, SECOND HARBOR ENGINEERING CO., LTD NO.5 BRANCH
Wuhan, Hubei
China
932850380@qq.com

2024 4th International Conference on Computer Science and Blockchain (CCSB) | 979-8-3503-5650-2/24/$31.00 ©2024 IEEE | DOI: 10.1109/CCSB63463.2024.10735538

***Abstract*—With the development and widespread application of IoT technology, there are problems affecting the performance of IoT databases, such as high dimensionality of database parameters and difficulty in parameter classification. This article focuses on the problem of optimizing data parameters in the Internet of Things. Using AI agent technology, an AI agent database parameter tuning method is proposed to achieve on-demand classification of IoT parameters and effectively expand tuning parameters. Through the collaboration and learning of multiple agents, reasonable parameter settings are recommended for IoT databases. Experiments were conducted in three workload environments: YCSB, TPC-C, and Seats. The results showed that compared to other mainstream models, the model proposed in this paper is more likely to expand the number of adjustable parameters for IoT databases and achieve better database performance.**

***Keyword-Internet of Things; Database management; AI Agent*.**

## I. INTRODUCTION

In the 21st century, human society has entered the information age and begun a new technological revolution. The main content of this technological revolution is the research on the Internet of Things [1-2]. The Internet of Things is an Internet way that can achieve balanced value distribution. It relies on cloud computing to integrate, manage, store and mine massive data in different data formats collected from all walks of life, and control these sensor networks. As the boundary of the Internet of Things, sensors are responsible for tasks such as data collection, data preprocessing, and data transmission.

AI Agent refers to an intelligent agent that can perceive the environment, autonomously understand, make decisions, and perform actions [3]. It has the ability to think independently, call tools, and gradually achieve given goals. According to the definition provided by ooldridge and Jennings in 1995, an AI agent is a computer system located in an environment that is capable of acting autonomously to achieve its design goals. AI agents should possess four basic attributes: autonomy, responsiveness, social ability [4], and proactivity. AI Agent technology can effectively achieve information exchange and collaborative work among multiple agents. AI Agent technology provides a new solution for complex control of the Internet of Things. Based on the advantages of AI Agent technology, such as autonomy, learning and optimization capabilities, distributed and collaborative control, this paper proposes an AI Agent based communication scheduling model for the Internet of Things, which improves the adaptability, stability, efficiency and performance of IoT systems through IoT data scheduling AI Agent technology.

## II. BASIC THEORY

### *A. AI-Agent theory*

AI agents have the following characteristics: autonomy: AI agents can make decisions and take actions autonomously without direct human intervention [5]. Adaptability: AI agents can adjust their behavior according to changes in the environment to adapt to different situations. Learning ability: AI agents can continuously improve their performance and adaptability through learning [6-7]. Interactivity: AI agents can interact with other agents or humans to complete tasks together.AI agents can be divided into reactive agents, cognitive agents, and hybrid agents. Reactive agents only make decisions and take actions based on the current environmental state, without memory and learning abilities. Cognitive agents have memory and learning abilities, and can make decisions and take actions based on past experience and knowledge.

### *B. AI Agent Reinforcement Learning Algorithm*

AI Agent reinforcement learning algorithm is a method that enables agents to learn optimal behavioral strategies by interacting with the environment [9-10]. The environment provides a reward signal after the agent takes action. The agent consists of a strategy function and a value function, with the goal of maximizing long-term cumulative rewards. Common algorithms include Q-learning, Deep Q-Network (DQN), and Policy Gradient Algorithm. This algorithm can be applied in fields such as robot control, gaming, and autonomous driving.

## III. DESIGN CONCEPT OF COMPLEX CONTROL SYSTEM FOR INTERNET OF THINGS BASED ON AI AGENT

### *A. Framework*

The basic design concept of the AI Agent framework is based on meeting the differentiated needs of intelligent application scenarios in the Internet of Things, integrating the latest technologies of AI big models and AI Agent application construction in the IT field. The architecture is shown in Figure 1. The specific design concept is as follows:

(1) Multi model fusion: The intelligent scenarios of the Internet of Things have different requirements for inference latency, interpretability, etc., and can meet the performance



Authorized licensed use limited to: CQUniversity. Downloaded on September 30,2025 at 18:27:34 UTC from IEEE Xplore. Restrictions apply.

requirements of multi scenario AI through the fusion of large and small models. Therefore, the base model can cover various technologies such as general LLM large models, professional IoT field large models, scenario based small models, knowledge graphs, etc.

(2) Customized Agent: AI Agent is designed for specific application scenarios in the Internet of Things field, such as network optimization, operation and maintenance, and perception business empowerment. In the framework system design, customized intelligent agent construction can be carried out by combining existing tools, expert experience, knowledge bases, etc. in specific IoT application fields, fully utilizing existing network knowledge to solve complex problems in IoT scenarios more quickly and efficiently.

(3) Internet of Things: AI agents interact with the environment in real-time to obtain changes in the environment, and execute tasks and adjust strategies based on this information to achieve expected goals. The external environment of IoT artificial intelligence agents includes the control layer, perception layer, data layer, application layer, etc. By implementing interaction between various layers of the Internet of Things and agents, the flexibility, scalability, and adaptability of the system can be improved, system complexity and maintenance costs can be reduced, and personalized needs in different scenarios can be bettermet.

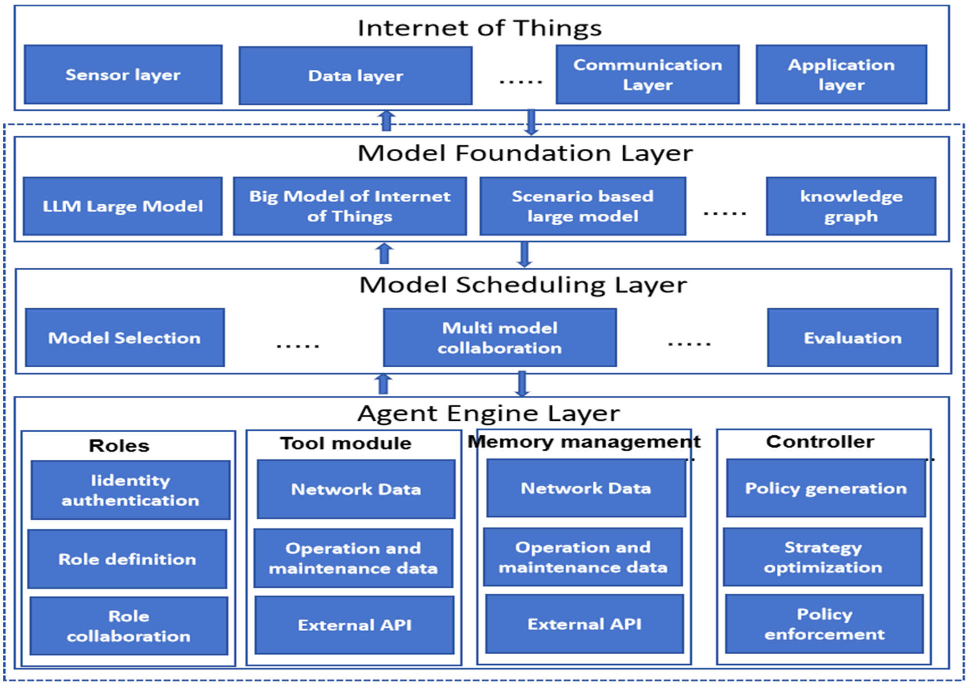


Figure 1. A Complex Control Framework for the Internet of Things Based on AI Agents

*B. An AI agent based data management model for the Internet of Things*

Research on AI agent application using IoT data management as an example.

1. Problem description

Definition 1 (IoT database parameter settings). Set the database parameter set to $P=\{p_1,p_2,...,p_n\}$, where each parameter consists of a parameter name, value range, and default value, denoted as *(Name$_i$, Vi, default$_i$)*. Parameters can be integer, floating-point, or enumeration types.

Definition 2 (Optimal Parameter Setting). Given a set of parameter settings $P$ and a workload $W$, the database satisfies formula (1) under $W$.

$$P^* = \arg\max f(P,W) \tag{1}$$

Among them, $f$ is the objective function, and $P^*$ is the optimal parameter setting under workload $W$.

2. Model building

The design concept of multi-agent reinforcement learning algorithm based on AI agent is as follows:

1) Assign a type of parameter, such as I/O parameter type, to each agent. Each agent only needs to explore the impact of the parameters they are responsible for on the performance of the Internet of Things, and optimize their own policy network based on the reward model, so that the parameters recommended by the policy network make the database performance better;

2) Multiple agents are trained through permutation and combination to explore the impact of the relationship between different parameter types on database performance;

3) Each agent optimizes high-dimensional database



Authorized licensed use limited to: CQUniversity. Downloaded on September 30,2025 at 18:27:34 UTC from IEEE Xplore. Restrictions apply.

parameters by tuning low dimensional synthesis parameters, indirectly improving database performance and greatly reducing the complexity of model training;

4) Using the performance metric throughput of the Internet of Things database as a feedback metric for the reward model, the reward for optimizing the model is calculated through the reward model, enabling the agent's policy network to obtain better optimization strategies.

The specific algorithm is as follows:

Algorithm 1: IoT Data Control Algorithm Based on AI Agent

**Input:** Number of joint training sessions M
**Output:** The updated network for each agent

FOR episode = 1 TO *M* DO
FOR I = 1TO N DO
According to the strategy network $\pi$ of Agent *i*, the action $a_t^i$ is obtained;
END FOR.
Arrange and combine to obtain a combination list of all combined actions, $action\_list$;
FOR *x* IN $action\_list$ DO
Execute action $a_t^x$ in the environment to obtain a new observation $o_t^x$ of the environment;
For *agent* IN *x*//Traverse each agent in tuple *x*
Computing environment rewards;
Store $(o_{t-1}^x, a_{t-1}^x, r_{t-1}^x, o_t^x)$ in the Agent's experience buffer;
END FOR.
END FOR.
END FOR.

Among them, $a_t^x$ represents the combined action of multiple agents, for example, $a_t^x = [a_t^1, a_t^3]$, represents $O_t^x = [o_t^1, o_t^3]$.(·) represents a set of experiences, including the environmental observation value $o_{t-1}^x$ at time *t-1*, the combined action $a_{t-1}^x$, the reward $r_{t-1}^x$, and the current environmental observation value $o_t^x$ at time t.

3. Detailed workflow of the model

1) Each agent generates an action information $a_t^i$ based on the policy network, which is the input of the low dimensional mapping model. The calculation method is shown in formula (2). Among them, $\mu$ represents the deterministic policy function, $\theta$ is the neural network parameter of the policy network, $o_{t-1}^i$ is the environmental observation value, $N_t$ represents the OU noise at time *t*, used to increase randomness.

$$a_t^i = \mu_\theta^i(o_{t-1}^i) + N_t \tag{2}$$

2) Mapping low dimensional synthetic parameter vectors to high-dimensional database parameter vectors.

3)Map high-dimensional database parameter vectors to specific database parameters.

4) Mapping high-dimensional IoT parameter vectors to specific data parameters. Due to the control of multiple agent tuning in this model, each agent will generate a subset of database parameter settings. This model uses a combination algorithm to arrange and combine the actions of intelligent agents. Taking the number of agents as 3 as an example, iterative training can generate 7 sets of recommended parameter settings S={1,2,3,[1,2],[1,3],[2,3],[1,2,3]} within one cycle.

5) Apply the parameter setting samples to the data management system in sequence.

6) Load test the recommended parameter settings and evaluate their effectiveness.

7) After workload testing, a set of experience data can be obtained, and then stored sequentially in the experience pool of each agent.

8) Each agent needs to be optimized. For an agent, four networks are applied: policy network π, target policy network Target $\pi$, value network Q, and target value network Target *Q*. The strategy network $\pi$ is mainly responsible for generating low dimensional synthesis parameters, and its objective function is shown in formula(3). Among them, *B* is the number of training samples, *k* represents the *k-th* sample, $\mu$ is the deterministic policy function, $O_{k-1}^x$ is the environmental observation value, which represents the performance data obtained by the Agent's participation in optimization, set x stores the Agent's ID, which uniquely identifies an Agent, $\theta$ is the neural network parameter of the policy network, and $\varphi$ is the neural network parameter of the value network.

$$\nabla_{\theta^i} J^i \approx \frac{1}{B}\sum_{k=1}^{B} \nabla_{\theta^i}\mu^i(O_{k-1}^x)\nabla_{a^i}Q(o_{k-1}^x, a_{k-1}^x \mid \varphi^i) \tag{3}$$

The update of the target policy network Target $\pi$ is shown in formula (4), where τ is the discount factor.

$$\theta'^i \leftarrow \tau\theta^i + (1-\tau)\theta'^i \tag{4}$$

The objective function of the value network is shown in formula (5):

$$L(\theta^i) = \frac{1}{B}\sum_{k=1}^{B} (y^i - Q(\hat{o}_{k-1}^x, \hat{a}_{k-1}^x \mid \varphi^i))^2 \tag{5}$$

Among them, $y^i$ is an evaluation value based on partial true values.

The update of the target value network Target Q is shown in formula (6):

$$\varphi'^i \leftarrow \tau\varphi^i + (1-\tau)\ \varphi'^i \tag{6}$$

The main function of the strategy network is to generate action information, and the value network evaluates and scores the action information. The higher the score, the better the strategy. Therefore, the value network can guide the optimization of the strategy network.



Authorized licensed use limited to: CQUniversity. Downloaded on September 30,2025 at 18:27:34 UTC from IEEE Xplore. Restrictions apply.

### C. Reward Model

This article uses throughput as an indicator for calculating rewards, which refers to the number of requests or transactions processed by the system within a certain period of time. In database systems, it typically refers to the number of queries or transactions processed per second. High throughput indicates that the system can effectively handle a large number of requests, which means that the system has good performance and efficiency. Firstly, a calculation method for the throughput change rate is provided, using time t-1 and time 0 as reference points to calculate the performance change rate from time t to the previous time t-1, as well as from time t to time 0. This is because a single performance change rate from time t to time t-1 cannot reflect the improvement in performance. The calculation method for throughput change rate is shown in formula (7).

$$\begin{cases} \Delta T_{t\to t-1} = \dfrac{T_t - T_{t-1}}{T_{t-1}} \\ \Delta T_{t\to 0} = \dfrac{T_t - T_0}{T_0} \end{cases} \tag{7}$$

The calculation of the reward r designed in this article is shown in formula (8).

$$r = \begin{cases} ((1+\Delta T_{t\to 0})^2 - 1)\left|1+\Delta T_{t\to t-1}\right|, \Delta T_{t\to 0} > 0 \\ (1-(1-\Delta T_{t\to 0})^2)\left|1+\Delta T_{t\to t-1}\right|, \Delta T_{t\to 0} \le 0 \end{cases} \tag{8}$$

## IV. Experimentation

### A. Experimental setup

Comparative models: Adopting mainstream and classic algorithms based on heuristic models OpenTurner, Bayesian based models CGPTuner, and attention based deep reinforcement learning models WATuning. The basic setup of the model in this article: Firstly, four agents are set up, responsible for optimizing query parameters, log parameters, checkpoint parameters, and hardware parameters, with each agent responsible for 20 parameters. The soft update parameter $\tau$ of the target network is 0.001, the discount factor $\gamma$ is 0.95, the batch sample size B is 32, and the soft update time period T is 5. Evaluation metric: Throughput is an evaluation metric for database performance, measured in reqs/sec; The evaluation index for the quality of a model is the performance improvement ratio: PIR (Performance Improvement Ratio)=average performance improvement÷ initial performance. Workload: Database performance testing uses the classic benchmark dataset YCSB, which is used to evaluate the performance of cloud based database storage systems; TPC-C is a classic transaction processing performance benchmark used to evaluate the performance of relational database systems in multi-user transaction processing environments; Seats is a simulated workload for IoT systems used to evaluate the performance of concurrent access and transaction processing. This workload is typically used to test the performance of IoT database systems in handling real-time transactions and concurrent user requests.

### B. Experimental Result

In order to investigate the database performance of the model in different workload scenarios, the throughput improvement of the model was statistically analyzed under three workloads: YCSB-B (95% read and 5% write), TPC-C, and Seats. The number of iterations for each model was set to 600, and Figure 2 shows the changes in database throughput under different workload tests. It can be found that the OpenTuner model based on heuristic methods performs the worst and is unstable. Using this method for tuning can only select the optimal solution from multiple tuning processes. Because it is based on heuristic methods, when there are too many parameters, it is usually not possible to obtain excellent parameter settings; The WATuning and CGPTuner models perform well, both of which can achieve good database performance after about 150 rounds of tuning; And the model in this article is significantly better than other methods. In the Seats test, around the 90th round, the model converged and obtained better parameter settings than other models. The experimental results show that our model performs better than other mainstream models in different workload scenarios.

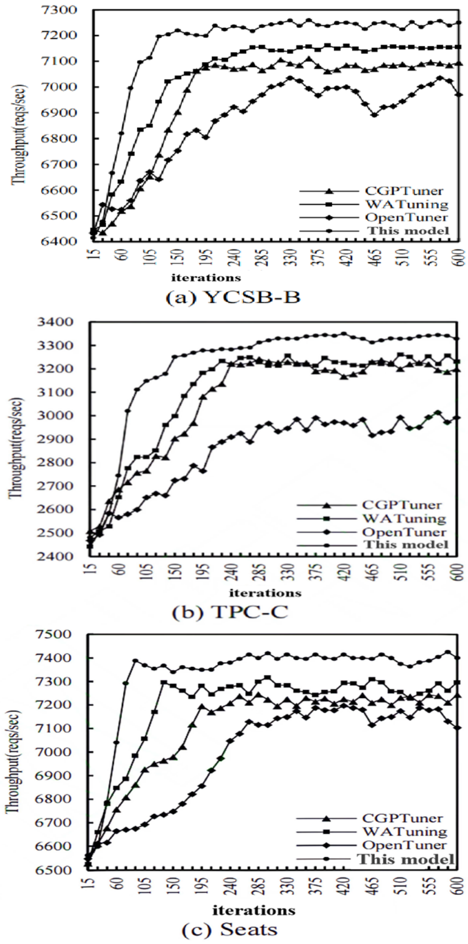


Figure 2. Comparison of throughput of different algorithms under different workload tests

## V. Conclusion

This article proposes an AI agent database parameter tuning method for optimizing IoT data parameters, aiming to achieve on-demand classification of IoT parameters and effectively expand tuning parameters. Through the collaboration and learning of multiple agents, reasonable parameter settings are



Authorized licensed use limited to: CQUniversity. Downloaded on September 30,2025 at 18:27:34 UTC from IEEE Xplore. Restrictions apply.

recommended for IoT databases. Experiments were conducted in three workload environments: YCSB, TPC-C, and Seats. The results showed that compared to other mainstream models, the model proposed in this paper is more likely to expand the number of adjustable parameters for IoT databases and achieve better database performance.

## REFERENCES


[1] Chen Xinyu, Wang Weibin, Lu Guanghui Framework and Application of 6G Endogenous Intelligence Technology Based on AI Agent [J] Mobile Communications, 2024, 48 (07): 28-32

[2] Wei Xueling Exploration of E-commerce Laboratory Construction Based on Enterprise Management Model AI Agent [J] Guangxi Education, 2024, (17): 105-109+120

[3] Lin Meirong Construction and Application of Translation AI Intelligent Agent Project Based on Large Language Model [J] Wireless Internet Technology, 2024, 21 (07): 42-45

[4] Ruan Runsheng Moving from Large Models to AI Agent Capital Deep Mining Computing Power Ecological Chain [N] Securities Times, February 6, 2024 (A05) DOI:10.38329/n.cnki.nzjsb.2024.000589.

[5] Zhang Zitong Is the next battle about AI agents moving towards practical implementation of large-scale models? [N]. 21st Century Business Herald, December 22, 2023 (010) DOI:10.28723/n.cnki.nsjbd.2023.005121.

[6] Liu Shanglin, Chen Yiduo, Li Ke Research on Collaborative Autonomous AI Agent System and Industrial System Network Security Application [J] Industrial Information Security, 2023, (03): 61-67

[7] Ge M. Review on the Application of Deep Learning Algorithms in Video Game AI Agent[C]// ITM Department, Illinois Institute of Technology. Proceedings of the 3rd International Conference on Signal Processing and Machine Learning (part1). School of Information Science and Technology, Northwest University; 2023: 6. DOI:10.26914/c.cnkihy.2023.112826.

[8] Zhang X, He Z, Zou L, et al. An Auxiliary Decision Method for Playing of the Poker2to1 Agent [C]//Northeastern University, Information Physics System Control and Decision Professional Committee of the Chinese Society of Automation Proceedings of the 34th China Conference on Control and Decision Making (1) School of Artificial Intelligence, Chongqing University of Technology; 2022: 5. DOI:10.26914/c.cnkihy.2022.020759.

[9] Li Chang, Gu Hanjie Design and Implementation of Football Competitive Games Driven by ML Agents [J] Fujian Computer, 2022, 38 (01): 81-84 DOI:10.16707/j.cnki.fjpc.2022.01.019.

[10] Liu Chengjie, Jin Xuemei, Wang Hongwei, etc Overall Design of Distributed AI Application in Air Traffic Control Simulation [C]//Chinese Command and Control Society Proceedings of the 9th China Command and Control Conference Nanjing Laisi Information Technology Co., Ltd; Civil Aviation Administration of China Operations Monitoring Center; 2021: 7. DOI:10.26914/c.cnkihy.2021.0112




Authorized licensed use limited to: CQUniversity. Downloaded on September 30,2025 at 18:27:34 UTC from IEEE Xplore. Restrictions apply.