SSRG International Journal of Electronics and Communication Engineering
ISSN: 2348-8549/ https://doi.org/10.14445/23488549/IJECE-V13I11P123

Volume 13 Issue 11, 374-385, November 2026
© 2026 Seventh Sense Research Group®

Original Article

Level-based Agentic AI for Security and Data-Performance Management in Distributed Fog-Cloud Environments

K Sindhuja[1], M Sathish Kumar[2]

[1,2]*Hindustan Institute of Technology and Science, Padur, Chennai, Tamilnadu, India.*

[1]*Corresponding Author: sindhujakesavan4@gmail.com*

Received: 30 September 2026 Revised: 29 October 2026 Accepted: 17 November 2026 Published: 27 November 2026

Abstract *- The increasing heterogeneity of security threats and data-processing requirements in distributed Fog–Cloud environments presents a need for more flexible mechanisms to ensure the provision of security and performance capabilities in accordance with specific requirements. Integrated Fog–Cloud frameworks normally incorporate security and performance mechanisms as predefined system-level functionalities, which hinders the ability to provide differentiated capabilities according to varying security and performance requirements. This paper develops a level-based agentic AI that creates specialized AI agents according to different levels of security and data-performance requirements. The Level-based Agentic AI maintains distinct categories of security agents and performance agents in which multiple agents are created based on the predefined capability levels. Security agents are designed to address different levels of requirements, from basic protection and monitoring to advanced threat detection and response. At the same time, performance agents address different levels of resource management, latency optimization, workload management and data-performance enhancement. When a Fog or Cloud server submits a requirement, the proposed Level-based Agentic AI analyzes the required levels of security, performance and identifies the appropriate agents among the agent population. The identified agents are then provisioned to the requested computational environment to carry out the required functionality. Thus, the proposed approach transforms the conventional system-level security and performance mechanisms into requirement-specific AI agent capabilities that are more flexible and intelligent. The Level-based Agentic AI is designed to improve security responsiveness, data-processing performance, resource utilization and adaptability under heterogeneous workloads and security conditions. Experimental evaluation illustrates the effectiveness of Level-based Agentic AI in improving security capability satisfaction, performance capability satisfaction, agent selection accuracy, security response time and resource allocation efficiency.*

Keywords *- Agent selection, Agentic AI, Agent provisioning, Fog-Cloud computing, Intelligent agent creation.*

1. Introduction

The growing number of Fog-Cloud distributed computing systems raises new challenges regarding the provision of proper security and persformance management in such heterogeneous systems. The workloads in Fog and Cloud servers are different, and the processing power and the security needs are distinct as well. Hence, flexible management solutions are needed to provide for the appropriate security and processing power. The selection and allocation of appropriate capabilities in accordance with the requirements of the individual computing system is essential.

The development of Artificial Intelligence (AI) technology has led to an increase in Agentic AI, which allows autonomous agents to reason, make decisions, execute tasks and interact with environment. The use of autonomous AI agents is an indicator of the increasing significance of agent intelligence in contemporary applications [1]. New advances in technology have also investigated secure Agentic AI and multimodal intelligence in the next generation of communication networks, reflecting the significance of autonomy and security [2]. Trust-aware orchestration involves means to measure agent reliability and coordinate autonomous agents based on needs [3]. Additionally, Agentic AI architecture and evaluation are significant in the development of autonomous agents that can execute tasks with minimal human interference [4].

The growing independence and connectivity of intelligent agents create various security issues, such as unwanted instructions, unauthenticated data, malicious interaction and undesirable behavior of the agents. These issues underline the significance of security assessment at the level of an individual agent and the need for suitable security

This is an open access article under the CC BY-NC-ND license (http://creativecommons.org/licenses/by-nc-nd/4.0/)

mechanisms [5]. Intelligent agents have been used in industries and energy systems for enabling optimization, resilience and adaptive decision-making [6]. In Distributed Agentic AI, there is the possibility of service orchestration in edge and cloud environments to facilitate the coordination of computing resources [7]. The use of Agentic AI in reliable system design indicates the capability of autonomous activity supported by Agentic AI [8]. Likewise, product lifecycle management through agent-based intelligence highlights the coordination of activities through autonomous intelligence [9].

In cases of distributed computing and communication scenarios, the application of Agentic AI can help with network monitoring, service orchestration, resource management and adaptive decision-making. Utilization of network telemetry combined with Agentic AI can be used as an option for achieving automatic network management, optimization and fault response [10]. Furthermore, there have been attempts at utilizing agent-based approaches towards security in order to protect and adapt in mobile enterprise settings [11]. Finally, decision-theoretic techniques represent an important basis for reasoning, decision-making and coordination in multi-agent systems [12].

Agent selection is another critical factor that needs to be considered in multi-agent settings since the selection of appropriate agents affect how the task is performed. Studies on reinforcement learning techniques have investigated agent selection and coordination based on observations from execution episodes [13]. The approach of algorithm selection can help agents select appropriate algorithms depending on the problem context and the amount of computational power available [14]. In a similar way, agent selection methods for federated learning also account for the amount of computation power in distributed wireless networks [15].

Despite the above advancements, there is a need for a more adaptable approach for the provision of differentiated capabilities in the context of security management and data performance management in distributed Fog–Cloud systems. The needs of security management could be related to the aspects of basic security and monitoring, advanced threat detection and mitigation. On the other hand, the needs of data performance management could be related to the issues of resource management, latency, workload management and data performance. Predefining the capabilities from the system perspective would limit the flexibility of assigning appropriate functionality based on the requirements of different environments.

To overcome the above difficulties, Level-based Agentic AI for Security and Data-Performance Management in Distributed Fog–Cloud Environments is proposed by this research. This involves the existence of distinct populations of agents for Security and Performance, where each population has distinct capability levels pre-defined. Security Agents cater to distinct capability levels related to various aspects of protection, monitoring, threat detection and response. Performance Agents cater to distinct capability levels related to various aspects of resource management, latency reduction, workload management and data-performance improvement. As soon as the Fog/Cloud server makes a request for the required level of security and performance, the Level-based Agentic AI determines the required level of security and performance and utilizes the LAGSP algorithm to select the appropriate agents from the populations.

The following are the major contributions of the study:

1. Level-based Agentic AI structure is set up based on different Security Agent and Performance Agent classes, where each class is defined based on predetermined capabilities.
2. Requirement analysis method finds out the security and performance capabilities that Fog and Cloud servers need to have.
3. The LAGSP mechanism selects and provisions appropriate agents according to the security and performance requirements analyzed.
4. The approach is experimentally validated based on security and performance capability satisfaction, agent selection accuracy, security response time and resource allocation efficiency.

2. Literature Review

Altintas et al. investigated the use of Artificial Intelligence (AI) agents for enabling perceptive, adaptive and anticipatory capabilities in 6G networks. The intelligent agents can support autonomous network sensing, decision-making, resource management and adaptation to changing communication conditions. Integrating AI agents provides a foundation for going beyond conventional connectivity to smart and self-adaptive 6G network architectures. However, the work provides a broad agent-enabled networking perspective and does not specifically address security-level or performance-level agent creation and selection in fog–cloud environments [16]. Altun et al. investigated the strategic implementation of super-agents in heterogeneous multi-agent training for improving adaptability in complex simulation environments. The approach shows how higher-level intelligent agents can coordinate heterogeneous agents and improve autonomous decision-making under changing operational conditions. The study emphasizes multi-agent coordination, adaptive learning and intelligent agent behavior in complex environments. However, the framework is designed for military simulation and does not specifically investigate agent provisioning, fog–cloud resource management, or security-oriented agent selection [17]. Adornetto et al. examined generative agents in agent-based modeling, focusing on application, validation and emerging challenges. The study highlights how generative AI can be

incorporated into autonomous agents that can produce context-dependent behaviors and interact within complex multi-agent environments. The work emphasizes agent intelligence, behavioral modeling, validation and the challenges associated with reliable generative-agent systems. However, it does not specifically address the creation and selection of specialized agents according to security and performance requirements in distributed computing environments [18].

Kostopoulos et al. investigated Agentic AI and its emerging applications, emphasizing autonomous agents that can plan, reason, decision-making and execute tasks with limited human intervention. The study discusses the transition from conventional AI models to agentic systems that can dynamically interact with environment and coordinate multiple activities. This provides an important foundation for developing intelligent agents that can have adaptive and autonomous operations. However, the work is mainly focused on education and does not specifically investigate Agentic AI for security and performance management in fog-cloud computing [19].

Yang et al. developed an intelligent agent-based prediction system with cloud computing for enterprise services in an IoT environment. The framework integrates intelligent agents, cloud computing and IoT data processing to support predictive decision-making and enterprise-service management. The approach shows that intelligent agents can utilize cloud resources for analyzing IoT information and providing adaptive predictions. However, the work does not explicitly consider multiple specialized agents for differentiated security and performance requirements and is based on an earlier generation of intelligent-agent architectures [20].

Mao et al. proposed a multi-agent reinforcement learning approach for capacity planning in vehicular fog computing. Multiple learning agents collaboratively determine computing-resource capacity according to changing vehicular workloads and service requirements. The approach shows how multi-agent reinforcement learning can provide adaptive resource provisioning and improve computing efficiency in dynamic fog environments. However, the framework primarily focuses on capacity planning and does not incorporate specialized security agents or security-performance-aware agent selection [21].

Hmimou et al. proposed a multi-agent system for cybersecurity threat detection and correlation using large language models. The framework employs multiple intelligent agents to analyze and correlate security information to enable more autonomous identification and interpretation of complex cybersecurity threats. The integration of LLMs with multi-agent cybersecurity provides a strong foundation for intelligent threat analysis and collaborative security decision-making. However, the framework does not explicitly create or provision agents according to different security levels and performance requirements within Fog-Cloud infrastructures [22].

Srichandan et al. proposed a secure and distributed placement approach for QoS-aware IoT requests in fog–cloud computing. The framework jointly considers security, distributed placement and Quality of Service (QoS) requirements when allocating IoT requests to available fog–cloud resources. This work shows the importance of incorporating security and performance requirements when making resource-placement decisions in heterogeneous fog–cloud environments. However, security and QoS are handled through the placement mechanism rather than through dynamically created specialized intelligent agents [23].

Mishra et al. proposed a federated deep Q-learning approach for collaborative computation and offloading of compute-intensive and latency-sensitive tasks in vehicular fog computing. The method combines federated learning with deep reinforcement learning to enable distributed task-offloading decisions while reducing the need for centralized model training. The approach addresses latency, computation requirements, task dependencies and resource utilization in dynamic fog environments. However, it primarily optimizes task offloading and does not provide intelligent agent creation based on differentiated security and performance requirements [24].

Suzuki et al. proposed a multi-agent deep reinforcement learning framework for cooperative computing offloading and route optimization in multi-cloud-edge networks. The framework enables multiple agents to collaboratively determine computing-offloading and routing decisions across distributed cloud and edge resources. The approach shows that multi-agent learning can be effective for adaptive resource allocation and cooperative decision-making in heterogeneous cloud-edge environments. However, the agents are primarily designed for computational optimization and routing rather than dynamically selecting specialized agents according to security and performance levels [25].

Mangalampalli et al. proposed a prioritized task-offloading mechanism for cloud–fog computing using an improved asynchronous advantage actor-critic algorithm. The framework prioritizes tasks according to requirements and dynamically determines suitable cloud or fog resources for execution. The reinforcement-learning mechanism enables adaptive task scheduling and resource utilization under changing workloads. However, the approach concentrates on task prioritization and offloading performance and does not explicitly incorporate security-level-based agents, performance-level-based agents, or intelligent agent provisioning [26].

Gu et al. developed a collaborative offloading method for digital-twin-enabled cloud-edge computing in the Internet of Vehicles. The approach combines digital twins with collaborative computing and offloading to improve resource utilization and task execution across distributed cloud-edge infrastructures. The framework shows how intelligent digital representations can support adaptive decisions in highly dynamic vehicular environments. However, the work focuses mainly on computation offloading and does not explicitly address autonomous security agents or differentiated security-performance agent selection [27].

Khan et al. proposed MAADRR, a Multi-Agent Adaptive Dynamic Round Robin approach for cloud scheduling using real cloud workload traces. The framework employs multiple agents to dynamically adapt scheduling decisions according to changing workload characteristics. The use of real cloud workload traces supports more realistic evaluation of adaptive resource scheduling and shows the potential of multi-agent mechanisms for cloud resource management. However, the approach primarily addresses workload scheduling and does not consider security-aware agent provisioning or the creation of specialized agents according to application requirements [28].

Mangiaracina et al. proposed an adaptive multi-agent-based approach for Data as a Service in fog computing. The framework uses multiple agents to support adaptive data-service management and decision-making within distributed fog environments. The approach shows how agent-based intelligence can improve service provisioning and resource utilization in fog computing while adapting to changing application requirements. However, the framework focuses on data-service adaptation and does not explicitly integrate security-level-based agent selection with performance-oriented agent provisioning [29].

Vijetha proposed an Agentic Intelligence framework for unified cyber defense across cloud, edge and IoT systems. The self-adaptive framework uses intelligent agents to support autonomous threat detection and coordinated cybersecurity operations across heterogeneous computing environments. This work is particularly relevant to Agentic AI, cybersecurity, cloud–edge–IoT integration, autonomous threat detection and adaptive security decision-making. However, it does not explicitly differentiate agents according to security levels and performance levels or provide a systematic mechanism for dynamically creating and provisioning specialized agents based on application requirements [30].

3. Comparative Discussion of Related Research

A comparative analysis of the discussed studies shows some distinctions in the use of techniques of agent coordination, security and performance management. In particular, the multi-agent capacity planning in Vehicular Fog Computing [21] is mostly related to the process of resource capacity evaluation, while the multi-agent approach to cybersecurity [22] is focused on threat detection and correlation. Likewise, secure QoS-aware Fog-Cloud placement [23] includes the processes of security and service quality evaluation during resource allocation without creating specialized agents with defined capabilities.

In turn, the multi-agent Cloud-Edge offloading and route optimization approach [25] is mainly concerned with computational performance rather than the requirement-based selection of specialized security and performance agents. Finally, the adaptive multi-agent data-service approach [29] is dedicated to data-service management and provisioning, while the Agentic Intelligence approach for cyber defense [30] is oriented at autonomous threat detection and security operations.

In contrast, Level-based Agentic AI uses distinct Security Agents and Performance Agents that are grouped based on distinct capability levels. The LAGSP process looks at the requirements provided by Fog and Cloud servers and recognizes what security and performance levels are needed, chooses the appropriate agents in the two groups and provisions them accordingly to the requesters' computational environment.

Thus, the key difference is the provisioning and organization of the specialized agents depending on the requirements of different security and data performance.

4. Research Gap

Research in Fog and Cloud environments so far has focused on security enhancement, resource management, privacy preservation and performance optimization using different types of AI techniques. However, there has been no arrangement in terms of these capabilities, as they are generally offered either as individual functions or at fixed system levels. In particular, the requests of Fog or Cloud environments are not considered in connection with the respective capability levels, which makes it unfeasible to assign them to a proper agent for execution.

Moreover, security management and data performance management are considered to be two separate processes, which creates a gap in terms of capability provisioning. The level-based agentic approach aims to overcome the identified deficiencies by utilizing separate Security Agent and Performance Agent populations in accordance with predetermined levels of capabilities.

The LAGSP mechanism is responsible for all the required steps, including analyzing the requirements, identifying the levels of security and performance, selecting agents and managing them for the respective Fog or Cloud environment.

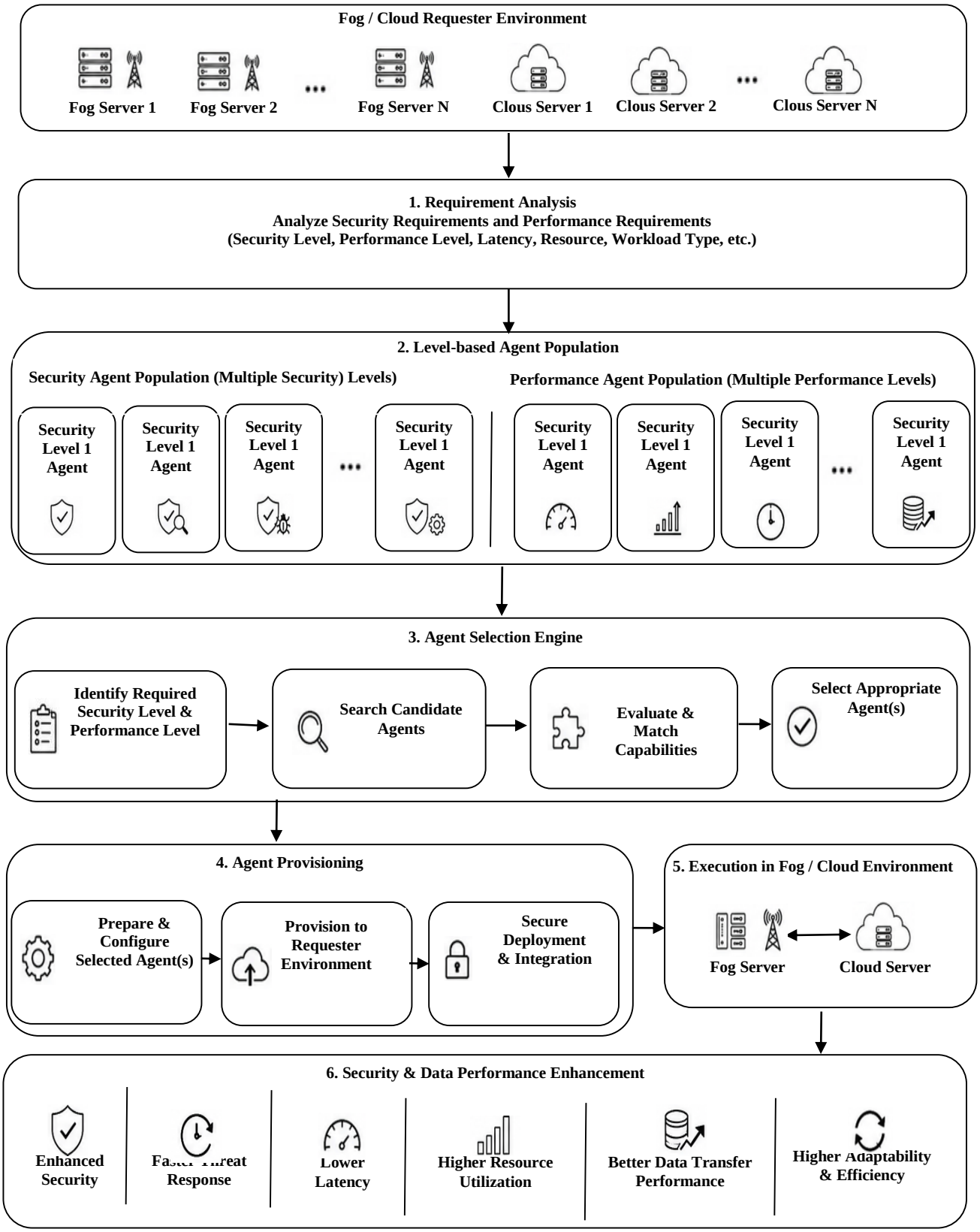

Fig. 1 Overall architecture of the level-based agentic AI framework for requirement-driven security and data-performance enhancement in fog–cloud environments

The architecture starts with the Fog and Cloud Requester Environment as shown in Figure 1. Here, either Fog or Cloud servers put forward security and data performance needs. This is analyzed by the Requirement Analysis module by checking factors such as security level requirements, performance level requirements, latency restrictions, resource requirements and load characteristics.

Then, based on the requirements identified, this framework utilizes two kinds of agent populations; namely, the Security Agent Population and Performance Agent Population. The Security Agent Population is made up of several agents that have been created for different security levels, and Performance Agent Population is made up of several agents that have been created for different performance levels. Security level agents have various capabilities ranging from basic security and monitoring to threat detection and handling, and performance level agents have resource management, latency improvement, workload management and data performance improvement capabilities.

The Agent Selection Engine selects candidates and assesses the abilities of those candidates in view of the sought-after level of security and performance, followed by the selection of suitable agents or combination of agents to be utilized within the requested environment. The selected agents pass through the Agent Provisioning stages in which they are set up, provisioned to the requested environment and connected to the Fog or Cloud infrastructure in a manner that secures its safety as well as coverage.

Following this, the agents that have been provisioned are made to fulfill required functions following the completion of provision of functions within the cloud network environment, which boosts security, performance and data transfer capabilities. The architecture in Figure below aims to achieve enhanced security, quicker response to threat detection, reduced latencies in usage of the resources and improved performance of data transfer.

Algorithm 1: Level-based Agent Generation, Selection and Provisioning
Input: Security levels, performance levels and requirements from a Fog and Cloud requester
Output: Appropriate security and performance agents provisioned to the requester
1. Start
2. Create Security Agent Repository
3. Create Performance Agent Repository
4. For each defined Security Level do
5. Identify the security capabilities required for that level
6. Create a Security Agent with those capabilities
7. Store the Security Agent in Security Agent Repository
8. End For
9. For each defined Performance Level do
10. Identify the performance capabilities required for that level
11. Create a Performance Agent with those capabilities
12. Store the Performance Agent in Performance Agent Repository
13. End For
14. Receive a requirement from Fog/Cloud requester
15. Analyze the security requirement
16. Identify the required Security Level
17. Analyze the data-performance requirement
18. Identify the required Performance Level
19. If there is a security capability required, then
20. Find the Security Agent for the required Security Level
21. Select the identified Security Agent
22. End If
23. If there is a performance capability required, then
24. Find the Performance Agent for the required Performance Level
25. Select the identified Performance Agent
26. End If
27. Configure the selected agent(s) according to the requester requirement
28. Provision the selected agent(s) to the Fog/Cloud requester
29. Activate the provisioned agent(s)
30. Execute the required security and/or performance functions
31. Evaluate the security and data-performance results
32. Measure Security Capability Satisfaction
33. Measure Performance Capability Satisfaction
34. Measure Agent Selection Accuracy
35. Measure Security Response Time
36. Measure Resource Allocation Efficiency
38. If the required security or performance level is not achieved, then
39. Analyze the requirement again
40. Identify the required level
41. Select the corresponding agent
42. Provision the selected agent
43. End If
44. Return the selected and provisioned agent(s)
45. End

The procedure of the Level-based Agent Generation, Selection and Provisioning (LAGSP) begins by generating separate agents for security and performance based on pre-determined capability levels. After a requested has been submitted by a Fog or Cloud client, LAGSP identifies the required security and performance levels, selects the appropriate agents and makes them available to the requester.

The selected agents now undertake respective tasks, which are assessed based on Security Capability Satisfaction, Performance Capability Satisfaction, Agent Selection Accuracy, Security Response Time and Resource Allocation Efficiency.

5. Dataset Description

Two freely available datasets have been used in the experimental evaluation process to illustrate the two most important requirements of the LAGSP framework, which include security requirements and data-performance requirements. The Cloud Access Control Parameter Management dataset is used to illustrate the security requirements, whereas the Multi-Cloud Service Composition Dataset has been used to illustrate data-performance requirements. The employment of two datasets ensures the independent evaluation of the security and performance aspects based on individual properties. The Cloud Access Control Parameter Management dataset illustrates the security parameters involved in the cloud access-control environment. This dataset contains information about authentication, authorization, identity management, access privileges, security policies, access control properties and others. All these parameters help in characterizing different degrees of security requirements. They can be employed to design and evaluate the performance of Security Agents based on capability levels. Thus, the dataset provides the security-oriented inputs to the requirement analysis and Security Agent selection within the LAGSP framework. Cloud Access Control Parameter Management - Kaggle

Multi-Cloud Service Composition Dataset is utilized for the indication of data-performance requirements in the setting of distributed cloud services. The dataset has information related to the service composition which can be used to perform analysis of computational and service-level requirements according to the multi-cloud execution. This information allows for the determination of different data-performance levels and the development of the corresponding performance agents. Thus, the dataset can be regarded as an input with regard to performance requirements analysis and the selection of performance agents for LAGSP. Multi-Cloud Service Composition Dataset - Kaggle

Together, the two sets of data provide input for the analysis from different aspects: Cloud Access Control Parameter Management is security-related, while Multi-Cloud Service Composition Dataset is data-performance-related. Such an approach allows LAGSP, first, to consider the requirements of each category separately, determine the level of capabilities that the agent should have to fulfill these requirements and assess the degree to which the selected agent satisfies the requirements of security and performance.

Two sets of data are taken as input to cover the two requirement categories of the Level-based Agentic AI architecture. Cloud Access Control Parameter Management provides the training data for security requirements, while Multi-Cloud Service Composition Dataset provides the training data for data-performance requirements. Both sets provide 1,000 instances each, making the pool of data for preparation and requirement generation consist of 2,000 instances. From the generated requirements, 250 security and 250 performance requirements are selected to form the set of requirements for experimentation, thus providing 500 requirements in total.

Table 1. Dataset configuration and requirement presparation

Dataset	Record Count Used	Purpose
Cloud Access Control Parameter Management	1,000	Generate security requirements
Multi-Cloud Service Composition Dataset	1,000	Generate performance requirements
Total Dataset Records	2,000	Preprocessing and requirement preparation
Security Test Requirements	250	Experimental evaluation
Performance Test Requirements	250	Experimental evaluation
Total Test Requirements	500	Final experiment

Table 1 displays the datasets and requirement configuration that were used for the evaluation. A total of 2,000 records have been prepared to create security and performance requirements, out of which 500 are chosen for the experimentation. The 250 security requirements and 250 performance requirements give an equal opportunity to evaluate the capabilities of both Security Agent and Performance Agent.

5.1. Level-based Security Agent Generation

Level-based security agent generation module generates multiple Security Agents based on various security capabilities levels. Security-related attributes from the Cloud Access Control Parameter Management dataset have been considered for characterizing requirements like authentication, authorization, identity management, access privileges and security policies. Normalized security attributes are then combined to calculate security capability scores for different levels, and based on this, Security Agents are arranged in accordance with security capabilities.

$$S_L = \frac{\sum_{i=1}^{n} w_i s_i}{\sum_{i=1}^{n} w_i} \quad (1)$$

Where Equation (1), S_L is the security capability value for a security level; s_i is the normalized value for the i-th security attribute; w_i is the weight of the same security attribute; and n is the total number of security attributes. Higher values of S_L indicate a higher security capability level. For a particular level of security, a Security Agent is created.

5.2. Level-based Performance Agent Generation and Requirement Matching

Multiple performance agents are developed through the Performance Agent Generation module, depending on the data performance level. Data and performance characteristics are derived from the Multi-Cloud Service Composition Dataset to assess specific computation and service performance needs for multi-cloud execution. Subsequently, the relevant performance indicators are adjusted and combined into a single performance capability level. Once a Fog or Cloud requester has specified its requirement, the calculated requirement level is compared against the capability level of available performance agents to identify the most suitable one.

$$P_L = \frac{\sum_{j=1}^{m} w_j \; p_j}{\sum_{j=1}^{m} w_j} \qquad (2)$$

Where, Equation (2) P_L is defined as the score of data-performance capability. Here, p_j is the percentage of the j-th performance criterion and denotes the value of the j-th performance parameter; w_j is the weightage of the performance criterion and denotes the significance of the j-th performance criterion; and m is the number of performance criteria. It is observed that the larger the value of P_L the higher the capability of data performance.

6. Experimental Analysis

The Level-based Agentic AI performance was assessed based on 500 test requirements that included 250 security ones and 250 data-performance requirements. The Level-Based Agentic AI is made of 100 agents: 50 Security agents and 50 Performance agents that consist of 5 security levels of 10 agents and 5 performance levels, respectively. The experiments were conducted in 10 runs to understand capability satisfaction, agent selection, response time and resource allocation based on different conditions of requirements.

6.1. Comparison of Performance

The results of Level-based Agentic AI have been compared with ERF-SHVO + ADGRU, followed by five measurement indicators. As per Table 2, the Level-based Agentic AI reached 96.80% Security Capability Satisfaction and 95.40% Performance Capability Satisfaction, while ERF-SHVO + ADGRU gained 82.40% and 79.60%, respectively. Level-based Agentic AI also achieved 98.86% Agent Selection Accuracy, which is more than the 97.10% acquired by ERF-SHVO + ADGRU. Regarding the execution efficiency, the Security Response Time has been decreased to 54.30 ms from 82.60 ms, and the Resource Allocation Efficiency has been improved from 78.60% to 88.40%.

The results displayed in Table 2 reveal that Level-based Agentic AI makes things better with relations to all its five evaluation parameters. Enhanced Capability Satisfaction with regards to security has increased by 14.40%, and improved Capability Satisfaction with regards to performance has surpassed the previous 15.80%. Agent Selection Accuracy has improved by 1.76%, which means that agents have been chosen more accurately in terms of required capability levels. Security Response Time has become 28.30 ms shorter (34.26% decrease), while Resource Allocation Efficiency has been enhanced by 9.80%. The listed results prove that proper arrangement of the Security Agents and Performance Agents in accordance with the level of capabilities makes it possible to better meet the requirements and, at the same time improves the security response and resource allocation efficiency.

From Figure 2, it is evident that Level-based Agentic AI has greater Security Capability Satisfaction of 96.80%, Performance Capability Satisfaction of 95.40%, Agent Selection Accuracy of 98.86% and Resource Allocation Efficiency of 88.40% compared to ERF-SHVO + ADGRU. It also has a reduced Security Response Time of 54.30 ms.

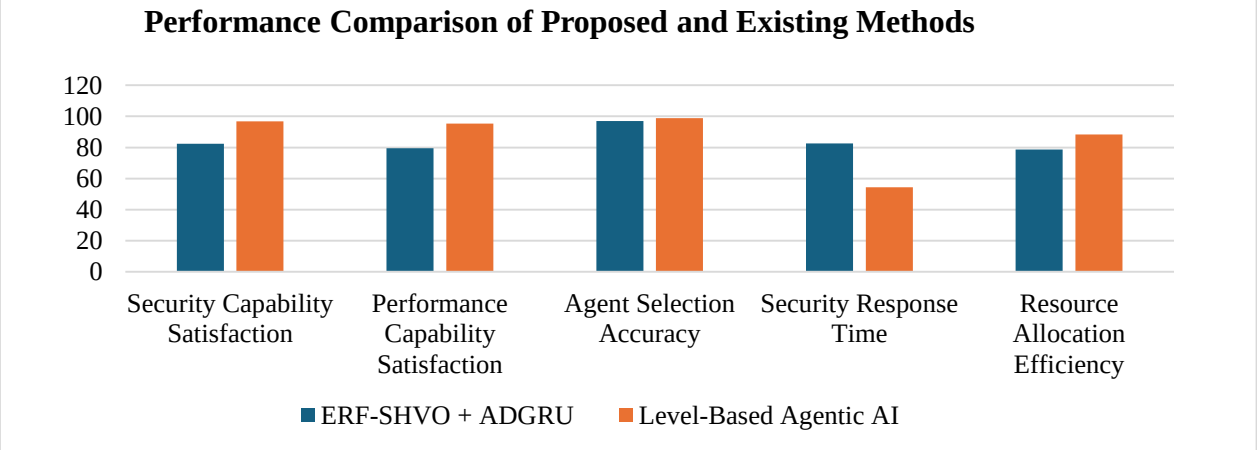

Fig. 2 Overall performance comparison between conventional agentic AI and level-based agentic AI

Table 2. Performance Comparison

Evaluation Metric	ERF-SHVO + ADGRU	Level-Based Agentic AI
Security Capability Satisfaction (%)	82.40	96.80
Performance Capability Satisfaction (%)	79.60	95.40
Agent Selection Accuracy (%)	97.10	98.86
Security Response Time (ms)	82.60	54.30
Resource Allocation Efficiency (%)	78.60	88.40

6.2. Reproducibility

For the purpose of reproducibility, the setup of Level-based Agentic AI experimentation includes constant agent population, capability levels, requirement workload and number of experiments conducted. These factors together make up the whole setup used for obtaining the performance results mentioned above.

Table 3 shows the fixed configuration that was used during the experiments. The number of agents within the agent population is 50 Security Agents and 50 Performance Agents. These 50 agents are distributed at 5 different levels. The number of requirements in the experimental process is 500 requirements. These 500 requirements consist of half security and half data performance requirements, and each experiment is repeated 10 times.

Table 3. Reproducibility configuration of level-based agentic AI

Parameter	Configuration
Programming Language	Java
Security Agents	50
Performance Agents	50
Total Agents	100
Security Levels	5
Performance Levels	5
Agents per Security Level	10
Agents per Performance Level	10
Security Requirements	250
Performance Requirements	250
Total Test Requirements	500
Experimental Runs	10

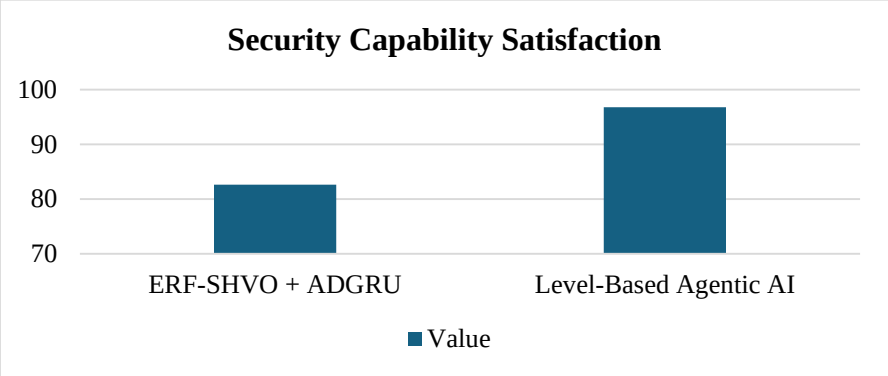

Fig. 3 Comparison of security capability satisfaction

Level-based Agentic AI satisfies 96.80% of the required security capabilities, while conventional Agentic AI satisfies only 82.40%. Figure 3 below shows these results.

Level-based Agentic AI has 95.40% satisfaction for performance capability, as can be seen in Figure 4, which is better than 79.60%.

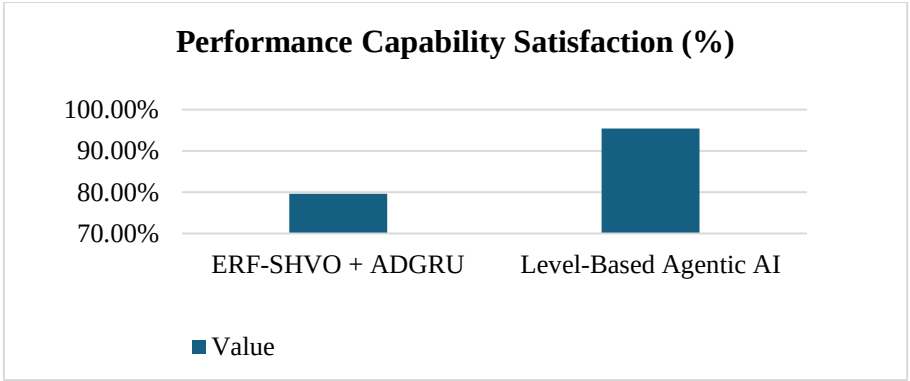

Fig. 4 Comparison of performance capability satisfaction

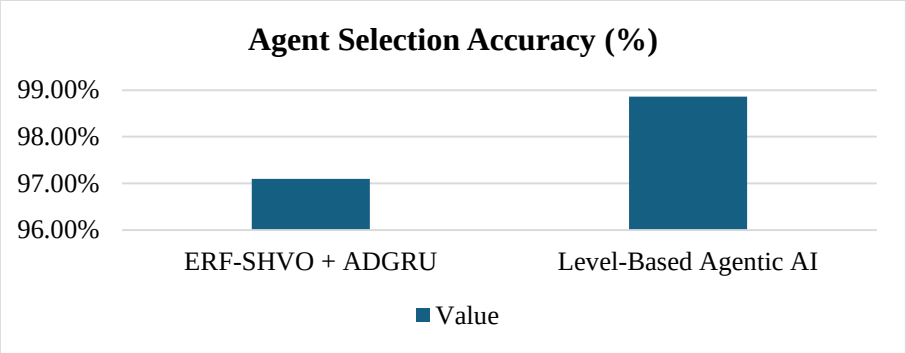

Fig. 5 Comparison of agent selection accuracy

Level-based Agentic AI achieves higher agent selection accuracy (98.86%) than ERF-SHVO + ADGRU (97.10%), as illustrated in Figure 5.

Level-based Agentic AI decreases the security response time of security by 82.60 ms to 54.30 ms, as depicted in Figure 6.

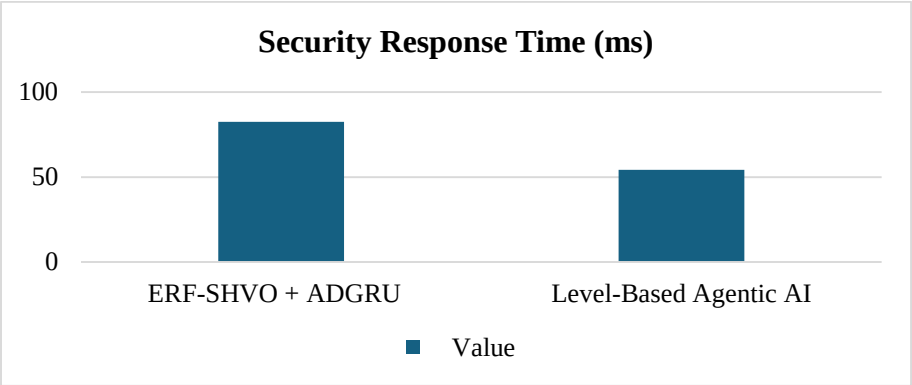

Fig. 6 Comparison of security response time

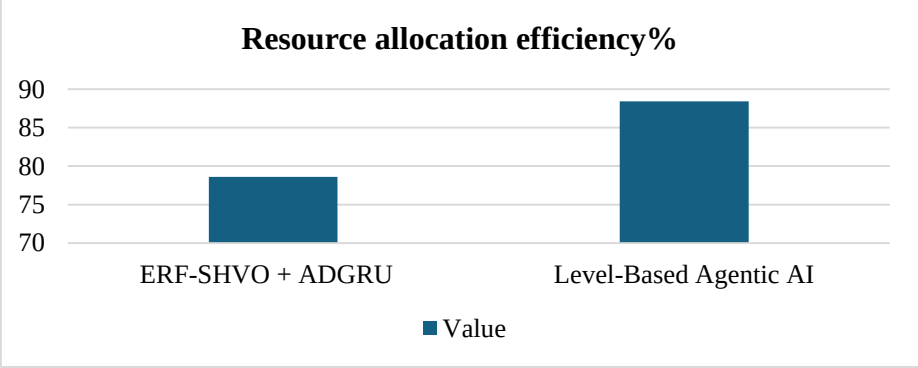

Fig. 7 Comparison of resource allocation efficiency

It is evident from Figure 7 that the resource allocation efficiency of Level-based Agentic AI is 88.40%, while that of conventional Agentic AI is 78.60%.

7. Results and Discussion

The findings of the experiment indicate that Level-Based Agentic AI is superior in terms of security and performance in data management compared to ERF-SHVO + ADGRU. The Security Capability Satisfaction increased from 82.40% to 96.80%, whereas the Performance Capability Satisfaction increased from 79.60% to 95.40%. The Agent Selection Accuracy also improved from 97.10% to 98.86%, showing accurate selection of appropriate agents according to the required requirements.

Moreover, the Security Response Time reduced from 82.60ms to 54.30ms, indicating an improvement in the speed of security processing. On the other hand, Resource Allocation Efficiency also improved from 78.60% to 88.40%. It can be seen that the level-based agentic AI

outperformed in terms of capability satisfaction, agent selection, security response and resource allocation. These performance gains are based on the capability level structure of Security Agents and Performance Agents.

LAGSP independently assesses security and performance needs and determines the capability levels corresponding to these needs before making any agent choices. This kind of selection based on requirements eliminates any mismatch of the requested capabilities and the chosen agents. In addition, using agent choice at predetermined levels leads to proper provisioning and contributes to better capability satisfaction and security response.

8. Conclusion

Level-based Agentic AI establishes a mechanism for addressing security and data-performance requirements in distributed Fog-Cloud environments. The architecture categorizes Security Agents and Performance Agents according to multiple capability levels, which enable identifying the required capability according to the submitted requirements. Such organization of agents allows for differentiated processing of security and performance demands, instead of utilizing the same processing mechanism for all requirements.

The experimental configuration was composed of 100 agents, including 50 Security Agents and 50 Performance Agents, which were divided into five security levels and five performance levels, respectively. The evaluation included 500 test requirements, of which 250 were security requirements and 250 were data-performance requirements, and the results illustrate the efficiency of the agent-selection mechanism. The results report on 96.80% SecurityCapability Satisfaction, 95.40% Performance Capability Satisfaction and 98.86% Agent Selection Accuracy. The efficiency results further report on a Security Response Time of 54.30 ms and 88.40% Resource Allocation Efficiency.

Overall, the results suggest that the utilization of specialized agents, which are organized according to capability levels, can enable more accurate requirement processing while also enabling faster security processing and efficient resource allocation. Compared to ERF-SHVO + ADGRU, the results suggest that Level-based Agentic AI can offer enhanced security capability satisfaction, performance capability satisfaction, agent selection accuracy, response time, and resource allocation efficiency.

To summarize, Level-based Agentic AI establishes a requirement-oriented mechanism for enhanced security and data-performance management in distributed Fog–Cloud environments. The categorization of agents according to multiple levels of capabilities enables differentiated processing of security and performance-related requirements while also enabling efficient resource allocation. The evaluation results further demonstrate the efficiency of the approach in terms of capability satisfaction, agent selection accuracy, sresponse time and resource allocation in comparison to related approaches.

Conflicts of Interest

The author(s) declare(s) that there is no conflict of interest regarding the publication of this paper.

Funding Statement

The author(s) received no specific funding for this research or publication.

References

[1] Marc Hasselwander, and Oliver Lah, “Agentic AI Arrives: How Gen Z Adopts Autonomous AI Agents,” *IEEE Access*, vol. 14, pp. 27083-27090, 2026. [CrossRef] [Google Scholar] [Publisher Link]

[2] Satyam Gaba et al., “Secure Agentic AI and Multimodal Intelligence in 6G Communication Systems,” *IEEE Open Journal of the Communications Society*, vol. 7, pp. 8926-8944, 2026. [CrossRef] [Google Scholar] [Publisher Link]

[3] Konstantinos I. Roumeliotis et al., “Agentic AI with Orchestrator-Agent Trust: A Modular Visual Classification Framework with Trust-Aware Orchestration and RAG-Based Reasoning,” *IEEE Access*, vol. 14, pp. 26965-26982, 2026. [CrossRef] [Google Scholar] [Publisher Link]

[4] Alaa Khamis et al., “Agentic AI Systems: Architecture and Evaluation using a Frictionless Parking Scenario,” *IEEE Access*, vol. 13, pp. 126052-126069, 2025. [CrossRef] [Google Scholar] [Publisher Link]

[5] Anshuman Chhabra et al., “Agentic AI Security: Threats, Defenses, Evaluation, and Open Challenges,” *IEEE Access*, vol. 14, pp. 49455-49482, 2026. [CrossRef] [Google Scholar] [Publisher Link]

[6] Agus Hasan et al., “Agentic AI in Wind Energy Systems: Multi-Agent Architectures for Optimization and Resilience,” *IEEE Access*, vol. 14, pp. 9935-9951, 2026. [CrossRef] [Google Scholar] [Publisher Link]

[7] Berend J. D. Gort, Godfrey M. Kibalya, and Angelos Antonopoulos, “Distributed Agentic AI Framework for Autonomous Edge-Cloud Service Orchestration,” *IEEE Transactions on Machine Learning in Communications and Networking*, vol. 4, pp. 1138-1157, 2026. [CrossRef] [Google Scholar] [Publisher Link]

[8] Lalit Narayan Mishra, and Biswaranjan Senapati, "Retail Resilience Engine: An Agentic AI Framework for Building Reliable Retail Systems With Test-Driven Development Approach," *IEEE Access*, vol. 13, pp. 50226-50243, 2025. [CrossRef] [Google Scholar] [Publisher Link]

[9] Artem Zaitsev, and Tatiana Kisel, "Digital Autonomation of Residential Construction Project Lifecycle Management via Agentic-AI-Driven PLM System," *Journal of Intelligent Construction*, vol. 4, no. 2, pp. 1-21, 2026. [CrossRef] [Google Scholar] [Publisher Link]

[10] Sergio Cruzes, "Telemetry and Agentic AI: Foundations for Optical Network Automation," *IEEE Access*, vol. 14, pp. 8800-8838, 2026. [CrossRef] [Google Scholar] [Publisher Link]

[11] Radhia Khellaf, and Souheila Boudouda, "Enhancing Mobile Enterprise Security: A Blockchain and Agent Paradigm-Based Approach for Continuous Protection and Rapid Adaptation," *IEEE Access*, vol. 12, pp. 108108-108120, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[12] Noushin Mazhar, and Maryam Kausar, "Rational Coordination in Cognitive Agents: A Decision-Theoretic Approach using ERMM," *IEEE Access*, vol. 11, pp. 92628-92646, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[13] Hyunsoo Lee et al., "Farthest Agent Selection with Episode-Wise Observations for Real-Time Multi-Agent Reinforcement Learning Applications," *IEEE Access*, vol. 12, pp. 108504-108514, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[14] Angelo Zapata, Julio Godoy, and Roberto Asín-Achá, "Anytime Automatic Algorithm Selection for the Multi-Agent Path Finding Problem," *IEEE Access*, vol. 12, pp. 62177-62188, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[15] Maria Raftopoulou et al., "Agent Selection Framework for Federated Learning in Resource-Constrained Wireless Networks," *IEEE Transactions on Machine Learning in Communications and Networking*, vol. 2, pp. 1265-1282, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[16] Mucahit Altintas et al., "Beyond Connectivity via Artificial Intelligence Agents Enabling Perceptive, Adaptive, and Anticipatory 6G Networks," *IEEE Access*, vol. 14, pp. 45977-46023, 2026. [CrossRef] [Google Scholar] [Publisher Link]

[17] Hüseyin Oktay Altun et al., "Strategic Implementation of Super-Agents in Heterogeneous Multi-Agent Training for Advanced Military Simulation Adaptability," *IEEE Access*, vol. 13, pp. 96544-96563, 2025. [CrossRef] [Google Scholar] [Publisher Link]

[18] Carlo Adornetto et al., "Generative Agents in Agent-Based Modeling: Overview, Validation, and Emerging Challenges," *IEEE Transactions on Artificial Intelligence*, vol. 6, no. 12, pp. 3165-3183, 2025. [CrossRef] [Google Scholar] [Publisher Link]

[19] Georgios Kostopoulos et al., "Agentic AI in Education: State of the Art and Future Directions," *IEEE Access*, vol. 13, pp. 177467-177491, 2025. [CrossRef] [Google Scholar] [Publisher Link]

[20] Weijun Yang et al., "Intelligent Agent-Based Predict System With Cloud Computing for Enterprise Service Platform in IoT Environment," *IEEE Access*, vol. 9, pp. 11843-11871, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[21] Wencan Mao et al., "Multi-Agent Reinforcement Learning-Based Capacity Planning for On-Demand Vehicular Fog Computing," *IEEE Transactions on Intelligent Vehicles*, vol. 10, no. 2, pp. 1043-1057, 2025. [CrossRef] [Google Scholar] [Publisher Link]

[22] Yasser Hmimou et al., "A Multi-Agent System for Cybersecurity Threat Detection and Correlation using Large Language Models," *IEEE Access*, vol. 13, pp. 150199-150215, 2025. [CrossRef] [Google Scholar] [Publisher Link]

[23] Suresh Kumar Srichandan et al., "A Secure and Distributed Placement for Quality of Service-Aware IoT Requests in Fog-Cloud of Things: A Novel Joint Algorithmic Approach," *IEEE Access*, vol. 12, pp. 56730-56748, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[24] Kaushik Mishra et al., "A Collaborative Computation and Offloading for Compute-Intensive and Latency-Sensitive Dependency-Aware Tasks in Dew-Enabled Vehicular Fog Computing: A Federated Deep Q-Learning Approach," *IEEE Transactions on Network and Service Management*, vol. 20, no. 4, pp. 4600-4614, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[25] Akito Suzuki, Masahiro Kobayashi, and Eiji Oki, "Multi-Agent Deep Reinforcement Learning for Cooperative Computing Offloading and Route Optimization in Multi Cloud-Edge Networks," *IEEE Transactions on Network and Service Management*, vol. 20, no. 4, pp. 4416-4434, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[26] S Sudheer Mangalampalli et al., "Prioritized Task Offloading Mechanism in Cloud-Fog Computing using Improved Asynchronous Advantage Actor Critic Algorithm," *IEEE Access*, vol. 12, pp. 136628-136656, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[27] Linjie Gu et al., "Collaborative Offloading Method for Digital Twin Empowered Cloud Edge Computing on Internet of Vehicles," *Tsinghua Science and Technology*, vol. 28, no. 3, pp. 433-451, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[28] Zafar Iqbal Khan, Muzafar Khan, and Syed Nasir Mehmood Shah, "MAADRR: Multi-Agent Adaptive Dynamic Round Robin for Cloud Scheduling using Real Cloud Workload Traces," *IEEE Access*, vol. 14, pp. 44165-44184, 2026. [CrossRef] [Google Scholar] [Publisher Link]

[29] Giulia Mangiaracina et al., "Efficient Data as a Service in Fog Computing: An Adaptive Multi-Agent based Approach," *IEEE Transactions on Cloud Computing*, vol. 11, no. 3, pp. 2646-2663, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[30] B. Vijetha, "Agentic Intelligence for Unified Cyber Defense: A Self-Adaptive Framework for Threat Detection across Cloud, Edge, and IoT Systems," *IEEE Access*, vol. 14, pp. 5104-5118, 2026. [CrossRef] [Google Scholar] [Publisher Link]