OPEN ACCESS
IOP Publishing | ISECAEP
Journal of Reliability Science and Engineering
J. Reliab. Sci. Eng. **1** (2025) 032001 (27pp)
https://doi.org/10.1088/3050-2454/adea7a

Topical Review

Reliability and security: from swarm robots to AI agents

Yuping Yan[1], Yuhan Xie[2], Junfeng Tang[1], Yuanshuai Li[3] and Yaochu Jin[1,*]

[1] Trustworthy and General Artificial Intelligence Laboratory, School of Engineering, Westlake University, Hangzhou, People's Republic of China
[2] School of Cyber Engineering, Xidian University, Xi'an, People's Republic of China
[3] School of Information Science and Technology, Nantong University, Nantong 226019, People's Republic of China

E-mail: jinyaochu@westlake.edu.cn, yanyuping@westlake.edu.cn, xieyuhan@westlake.edu.cn, tangjunfeng@westlake.edu.cn and liyuanshuai@westlake.edu.cn

Received 7 March 2025, revised 1 June 2025
Accepted for publication 1 July 2025
Published 11 July 2025

Abstract

Swarm robotic systems and AI agent systems are increasingly integrated into real-world applications, driving breakthroughs in autonomous traffic control, military surveillance, and agricultural monitoring. However, their deepening presence in critical infrastructure raises serious security and privacy concerns. Although both are built on shared principles of distributed intelligence and system architecture, a comprehensive understanding of their security threats and defense mechanisms is still lacking. This paper presents the first systematic survey that examines security vulnerabilities and countermeasures across both domains. We classify threats into three critical layers: physical, communication, and application, and provide an in-depth analysis of attack vectors along with corresponding mitigation strategies. Additionally, we explore the similarities and differences between security strategies in swarm robotics systems and AI agent systems, identifying opportunities to transfer valuable security insights between the two fields. Our findings highlight key research challenges, including real-time communication requirements, trade-offs between security, privacy, and efficiency, and challenges introduced by large language models. Finally, by outlining future research directions, this work aims to advance the development of secure and trustworthy intelligent systems.

Keywords: swarm robotic systems, AI agent systems, security, privacy, embodied intelligence, large language models

* Author to whom any correspondence should be addressed.

1. Introduction

Extensive research has shown that swarm behaviors in nature give rise to a form of collective intelligence, commonly referred to as **swarm intelligence** [1]. Swarm robotics leverages this collective intelligence, which emerges from the self-organization and distributed collaboration of numerous individual agents, enabling the resolution of complex problems without the need for centralized control [2]. These capabilities make swarm robotics highly applicable across various fields, including robotics, drone formations, and traffic management.

The rapid advancements in machine learning and large language models (LLMs), especially their capabilities in logical reasoning [3] and task planning [4], have shifted research focus from individual robots to more intelligent AI agent systems. Based on the definitions from Google's 2025 whitepaper [5], **intelligent AI agent systems** are autonomous entities that perceive their environment, make decisions, and take actions to achieve specific goals. These agents leverage AI techniques such as machine learning, reinforcement learning, and planning to adapt to dynamic conditions and optimize outcomes. By integrating large foundational models, these advanced systems can process both environmental and non-environmental data, enabling them to generate meaningful and context-aware actions. Another related concept is **embodied intelligence systems** [6]. These are AI-driven physical systems that engage with the real world through sensors, actuators, and adaptive behaviors. In summary, while AI agent systems serve as the 'brain,' embodied intelligence systems provide the 'body', enabling intelligent agents to interact with and adapt to the physical world.

The main difference between swarm systems and AI agent systems lies in their complexity: swarm systems consist of simple individual agents, whereas AI agent systems involve intelligent agents with more advanced capabilities. Despite their differences in functionality and intelligence levels, both face similar security challenges, such as fault tolerance, communication security, and interaction safety. These issues can lead to unintended system behavior and pose risks to task completion. Additionally, while data sharing and collaboration are fundamental to multi-agent systems, they also introduce significant privacy risks [7].

Recent advancements in reliability and security have made both swarm robotic systems and AI agent systems increasingly viable for real-world deployment. For example, H2 Clipper Inc. plans to implement safely operated robotic swarms in aerospace manufacturing [8]. Research presents the ethical governance analysis of swarm robotic systems in the real world [9]. In the AI domain, Microsoft introduced Security Copilot agents in March 2025 [10] to autonomously assist with critical tasks such as phishing detection, data protection, and identity management. Similarly, Google Cloud has launched AI-powered security agents as part of a unified security platform designed to streamline operations, triage, and threat intelligence [11]. Meanwhile, researchers have also highlighted AI security and cybersecurity risks in Internet of Things (IoT) devices, including drones and robots [12]. Additionally, they have pointed out cybersecurity threats associated with new software Bills of Materials that incorporate AI components [13]. However, a fundamental question remains: **how can swarm robotic systems and AI agent systems be designed with built-in security and privacy protections from the outset**?

In recent years, an increasing number of surveys have explored the security and trustworthiness of swarm robotic systems and AI agent systems, as shown in table 1.

Within the field of swarm robotic systems, researchers have proposed various frameworks and methodologies to enhance security and reliability. Hunt *et al* [14] introduced a safety checklist that addresses key factors such as ethics, legality, accountability mechanisms, and human–swarm interactions. While this checklist serves as a foundational framework for security research and practical applications in swarm intelligence systems, it primarily outlines critical issues in the form of questions without offering an in-depth analysis of current developments or specific technical solutions. Similarly, study [15] was among the first to systematically examine security challenges in swarm robotics, exploring multiple dimensions, including resource constraints, physical interference, control, communication, authentication, and key management. Although this survey effectively categorizes security threats, it lacks an in-depth discussion of defense mechanisms and the latest advancements in the field. Paper [16] presents a comprehensive discussion on performance, scalability, robustness, and adaptability in swarm intelligence systems, highlighting their role in building trust. However, it offers only limited insights into security and privacy protection and does not fully address recent technological developments.

In terms of AI agent systems, Xi *et al* [18] introduce a general framework for LLM agents, structured around three core components: brain, action, and perception. Beyond this, they explore applications in single-agent, multi-agent, and human-agent collaborations, as well as agent societies. However, it primarily focuses on the framework's design and functionality, with less emphasis on the critical aspects of reliability, robustness, and security in real-world deployments. Study [7] presents a comprehensive review of LLM-based agents, covering key topics such as cooperation paradigms, security challenges, privacy concerns, and future research directions. However, while their review discusses security at a conceptual level, it lacks an in-depth analysis of system vulnerabilities, attack vectors, and mitigation strategies, leaving gaps in understanding how to build resilient AI agent architectures. Neupane *et al* [19] provide a thorough investigation into security considerations in AI-driven robotics and offer a detailed taxonomy spanning three critical dimensions: attack surfaces, ethical and legal challenges, and human-robot interaction. However, despite its comprehensive threat analysis, the study primarily enumerates risks and vulnerabilities rather than offering concrete countermeasures, risk mitigation strategies, or adaptive security solutions, which are essential for ensuring long-term system robustness.

Compared to existing work, our paper provides the first comprehensive survey that systematically examines threats and countermeasures in both swarm robotic systems and AI agent systems. Additionally, we analyze the similarities and differences in security strategies and identify opportunities to

Table 1. Related survey.

Domain	References	Key points	Limitation
Swarm robotic systems	[14]	Provides a foundational safety checklist addressing ethics, legality, accountability mechanisms, and human–swarm interactions.	Primarily lists issues in question form without in-depth analysis or technical solutions for security challenges.
	[15]	Systematically examines security challenges across multiple dimensions: resource constraints, physical interference, control, communication, authentication, and key management.	Lacks detailed discussion on defense mechanisms and the latest advancements in security techniques.
	[16]	Provides a comprehensive discussion on performance, scalability, robustness, and adaptability, highlighting trust-building aspects in swarm intelligence systems.	Limited insights into security and privacy protection, and does not address recent technological developments in swarm robotics security.
	[17]	Provides a comprehensive research on Generative Artificial Intelligence (GenAI) in enhancing the trustworthiness, reliability, and security of autonomous systems.	Limited in terms of physical threats and countermeasures, as it primarily focuses on the AI aspects.
AI agent systems	[18]	Introduces a general framework for LLM agents, focusing on core components like brain, action, and perception. Explores applications in various collaboration models.	Focuses more on the framework's design and functionality, with less emphasis on security, reliability, and robustness in real-world deployments.
	[7]	Provides a comprehensive review of LLM-based agents, covering cooperation paradigms, security challenges, privacy concerns, and future research directions.	Lacks in-depth analysis of system vulnerabilities, attack vectors, and mitigation strategies, leaving gaps in building resilient AI agent architectures.
	[19]	Investigates security considerations across three dimensions: attack surfaces, ethical/legal challenges, and human-robot interaction. Offers a detailed taxonomy.	Primarily enumerates threats and vulnerabilities without providing concrete countermeasures, risk mitigation strategies, or adaptive security solutions.

transfer valuable security insights between these two fields. The main contributions of this paper are as follows:

- This paper introduces a detailed system evaluation framework that compares the structure and workflow of swarm robotic systems and AI agent systems, categorizing them into physical, communication, and application layers.
- We conduct an in-depth survey of security and privacy threats of these three layers for both systems, identifying common vulnerabilities as well as those unique to each domain. Corresponding countermeasures are proposed for both systems, with an evaluation of their effectiveness and limitations.
- We analyze the similarities and differences in attack strategies and defense mechanisms between swarm robotic systems and AI agents and outline future research directions aimed at enhancing the security, robustness, and reliability of intelligent systems.

The structure of the paper is as follows: Section 2 offers a comprehensive overview of both swarm robotic systems and AI agent systems. In Section 3, we delve into the security and privacy threats associated with these systems. Section 4 is dedicated to examining the countermeasures implemented to address these challenges. Moving forward, Section 5 provides a discussion on the key challenges and outlines future directions for research in this area. Finally, Section 6 wraps up the survey, presenting the concluding remarks and takeaways.

2. Overview from swarm robots to AI agents

In this section, we provide a comprehensive overview of swarm robotic systems and AI agent systems, including their definitions, characteristics, system structures, working pipelines, application scenarios and their related concepts. A detailed comparison can be found in table 2.

2.1. Overview of swarm robotic system

2.1.1. General structure. A swarm robotic system is a distributed system that emulates the self-organization and collaborative behaviors of biological populations in nature to accomplish complex tasks [20]. For instance, behaviors such as ant foraging, bee pollination, fish schooling, and bird migration. These systems consist of numerous simple individuals, each adhering to specific rules and interacting locally with others, collectively achieving highly complex and intelligent group behaviors globally.

Swarm robotic systems have three main characteristics:

- **Decentralized control mechanism**: the system operates without a central controller. Each agent independently makes decisions based on locally available information and simple behavioral rules.
- **High self-organization**: individuals spontaneously form global communication mechanisms through local interactions, without external intervention or global planning.

Table 2. Comparison of swarm robotic system and AI agent system.

Aspect	Swarm robotic system	AI agent system
Individual intelligence	Simple, rule-based; local perception only	Highly intelligent (e.g., LLMs, multimodal models); capable of reasoning, planning, and autonomous decisions
Communication modalities	Primarily wireless signals for local interaction	Include language, visual, audio, action-based, and symbolic communication
Working pipeline	More hardware-driven, relies on sensor-actuator interactions and task decomposition based on physical rules	More model-driven, leverages inference, logic, planning, and environment modeling for dynamic decision-making
Complexity	Emergent complexity from simple agents	Inherent complexity due to sophisticated individual models and interactions
Real-world applications	Focuses on military information gathering and mission support, maritime and deep-sea applications	Focuses on high-level reasoning, multimodal perception, knowledge manipulation, and learning

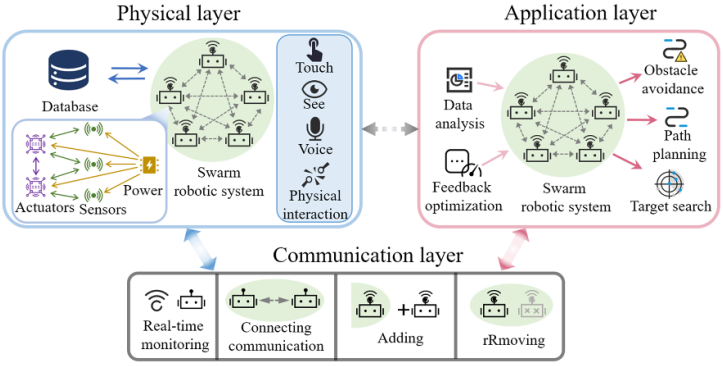

Figure 1. General structure of swarm robotic system.

- **Strong robustness**: the system's functionality is maintained despite the failure of individual agents, ensuring resilience and reducing the likelihood of systemic breakdown.

Swarm intelligence algorithms are integral to swarm robotic systems. They not only simulate self-organizing behaviors, such as employing gene regulatory network models to replicate the dynamics of multi-robot coordination [21], but also address key challenges in optimization, path planning, and task allocation. Drawing inspiration from collective behaviors in nature, a variety of swarm intelligence algorithms have been developed and applied across domains. For example, the ant colony optimization algorithm, proposed by Dorigo *et al* in 1992 [22], is commonly used for path planning; the particle swarm optimization algorithm, introduced by Kennedy *et al* in 1995 [23], is often used for multi-objective optimization; and the grey wolf optimizer, put forward by Mirjalili *et al* in 2014 [24], is frequently applied for feature selection in neural networks.

Based on decentralized control mechanisms and self-organization principles, the general structure of swarm robotic systems is depicted in figure 1. Structured hierarchically by functionality, it consists of three layers: the physical layer, the communication layer, and the application layer. The following sections provide a detailed elaboration on each layer.

- **Physical layer**: this layer provides the hardware infrastructure for swarm robotic systems, which is responsible for data acquisition and supporting physical interactions within and outside the system. It encompasses a wide variety of hardware components in large quantities, such as sensor modules, actuator modules, and power management modules.
- **Communication layer**: this layer facilitates information transmission and interaction among individuals in swarm robotic systems and manages the communication network topology formed during system operation. Network topology management includes real-time monitoring of individual states, managing communication connections, and handling the addition and removal of individuals. Wireless communication is commonly employed to transmit control commands and feedback.
- **Application layer**: this layer executes the predefined tasks of swarm robotic systems according to real-world production and living needs, assisted by data analysis and feedback optimization technologies. It includes tasks such as real-time obstacle avoidance, dynamic path planning, and target search.

2.1.2. Working pipeline. The operational pipeline of a swarm robotic system aligns closely with its general architecture. Upon receiving a task, the application layer analyzes and decomposes it, assigning roles and subtasks to individual agents. The communication layer transmits these assignments via wireless protocols, manages state monitoring, and coordinates information exchange. The physical layer enables robots to perform subtasks using sensors and actuators

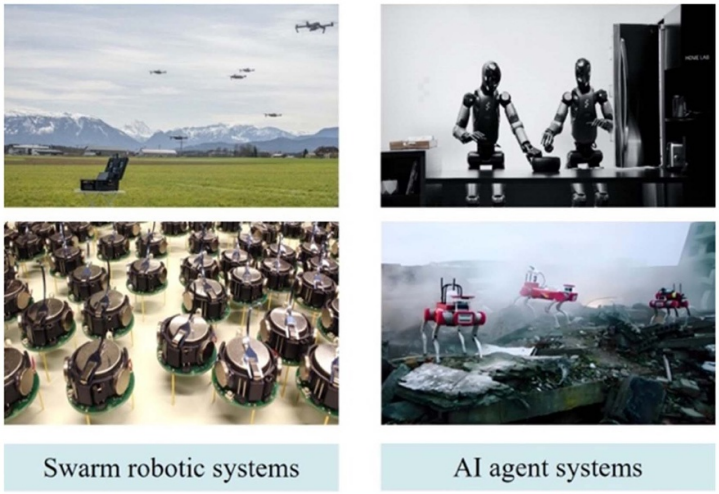

Figure 2. Applications of swarm robotic systems and AI agent systems.

while providing real-time feedback. Once all subtasks are completed, the communication layer consolidates the results, and the application layer applies optimization techniques before submitting the final outcome to the task issuer.

2.1.3. Real-world applications. Swarm robotic systems support a broad range of applications, as shown in figure 2 [25, 26]. For example, in **drone swarms**, swarm robotic systems enable drones to achieve dynamic path planning and real-time obstacle avoidance through information sharing and local perception, significantly enhancing the efficiency of post-disaster rescue and monitoring [27]. In **robot clusters**, swarm coordination improves collaborative operations and task optimization, with applications in industrial manufacturing, warehouse logistics, and agriculture [28]. In **intelligent transportation systems**, swarm robotic systems implement adaptive traffic signal control to effectively manage traffic flow and reduce congestion [29]. These examples highlight the pivotal role of swarm robotic systems in advancing digital transformation and intelligent automation across various sectors.

2.2. Overview of AI agent systems

2.2.1. General structure. An AI agent system consists of multiple highly intelligent agents that collaborate through logical reasoning, task planning, and local interactions to accomplish complex tasks in physical, virtual, or mixed-reality environments [29]. Each AI agent is a model-driven intelligent entity designed to generate appropriate responses based on input, enabling it to effectively achieve its objectives. Depending on different application scenarios, AI agents can be categorized as: Embodied Agents (including Action Agents and Interactive Agents), Simulation and Environment Agents, Knowledge and Inference Agents (including Knowledge Agents, Logic Agents, Agents for Emotional Reasoning, Neuro-Symbolic Agents), and Generative Agents such as AR/VR/mixed-reality Agents.

AI agent systems have the following four main characteristics:

- **Intelligence**: each AI agent integrates large models, such as LLMs, VLMs, and multimodal models, enabling autonomous planning, decision-making, control, and reasoning.
- **Flexible control mechanism**: these systems support both centralized and decentralized control. In centralized systems, a unified model (or 'shared brain') governs all agents, as exemplified by Figure's Helix robots [30]. In decentralized control mechanism, agents operate under separate models, relying on local perception and interactions to complete tasks.
- **Complexity**: AI agent systems demonstrate complex dynamics in response to environmental observations. They exceed traditional multi-agent systems in complexity, tackling tasks that surpass the capabilities of any single agent despite each agent's high level of autonomy.
- **Strong reliability**: each agent is capable of executing complex tasks independently, enhancing adaptability in dynamic environments. The system maintains resilience by allowing remaining agents to reallocate tasks when individual agents fail.

The general structure of AI agent systems, depicted in figure 3, comprises three hierarchical layers: the physical layer, the communication layer, and the application layer. Each layer fulfills distinct functions and interacts with adjacent layers to ensure effective system operation and task execution.

- **Physical layer**: this layer provides the hardware infrastructure for AI agent systems, responsible for data storage, providing computing power and system computational support, storing and invoking relevant large models, implementing possible mechanical structures (optional), and facilitating interactions with the environment.
- **Communication layer**: this layer facilitates information perception, transmission, and interaction among AI agents, while managing the dynamic communication network. It integrates visual (video/images), linguistic (text/knowledge), behavioral (action/cognition), and audio data

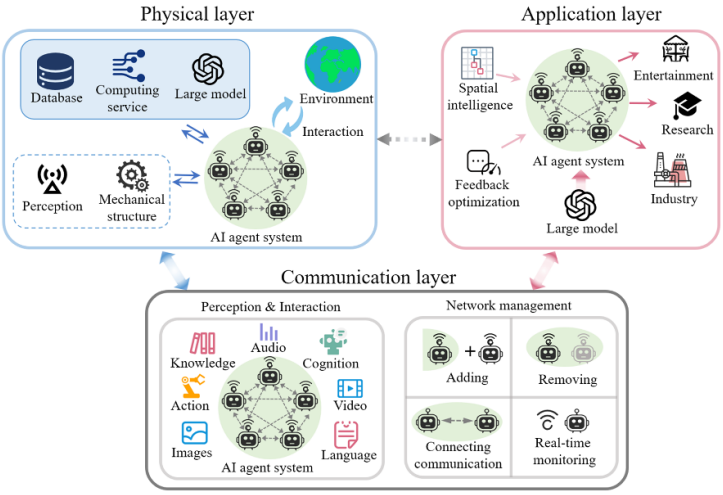

Figure 3. A typical structure of AI agent systems.

streams for perception and interaction. It also oversees real-time agent state monitoring, connection management, and dynamic agent integration.

- **Application layer**: this layer executes the predefined tasks based on real-world requirements, leveraging the large model technologies alongside spatial intelligence and feedback optimization techniques. It supports applications across domains such as entertainment, research, and industry.

2.2.2. Working pipeline. The working pipeline of AI agent systems spans their hierarchical structure, exhibiting key differences from swarm robotic systems. It begins at the application layer, where spatial intelligence techniques are used to interpret task requirements, decompose them into subtasks, and assign roles to individual agents while monitoring execution. The communication layer supports agent perception, inter-agent information exchange, and system state management. At the physical layer, agents access data, perform inference and planning using large models, and utilize computational resources to execute subtasks in the environment. Execution results are relayed back through the communication layer, aggregated, and optimized at the application layer before final submission to the task issuer.

2.2.3. Real-world applications. AI agent systems are applied across diverse domains, as illustrated in figure 2 [30, 31]. For example, in **robotics**, they enable human-machine collaboration for tasks like manufacturing and assembly processes [29]. In **gaming**, they create immersive virtual environments through realistic interactions between players and AI-driven characters [29]. In **healthcare**, AI agent systems can assist in medical diagnosis by integrating visual, textual, and sensor data, providing more accurate and timely insights to healthcare professionals [29]. These application scenarios highlight the revolutionary impact of AI agent systems on the intelligent development across various fields.

2.3. Comparisons of different concepts

Compared to swarm robotic systems and multi-agent systems [32], AI agent systems are distinguished by individuals with integrated model intelligence, possessing the highest degree of autonomy and reasoning. Additionally, they also support a more flexible control mechanism, which can adopt either centralized or decentralized.

Compared to agent AI systems [33], which consist of a single AI agent, AI agent systems comprise multiple agents, requiring more sophisticated integration across the physical, communication, and application layers. This enables them to tackle more demanding real-world and virtual tasks.

3. Security and privacy threats of swarm robots and AI agents

In this section, we describe the security and privacy threats faced by swarm robots and AI agents.

3.1. Security threats

Security threats of swarm robots and AI agents include physical attacks, communication attacks, and digital attacks. Specifically, we categorize each attack into three groups: common attacks, those unique to swarm robots, and those unique to AI agents. The primary categories of security threats are illustrated in figure 4. From the summary figure, we can observe that physical and communication attacks are largely similar in both systems, whereas AI agent systems are susceptible to a broader range of attacks—particularly in the domain of digital threats.

3.1.1. Physical attack. For both swarm robots and AI agents, physical attacks refer to behaviors that disrupt the normal operation of the system by interfering with the system hardware, communication, perception, decision-making

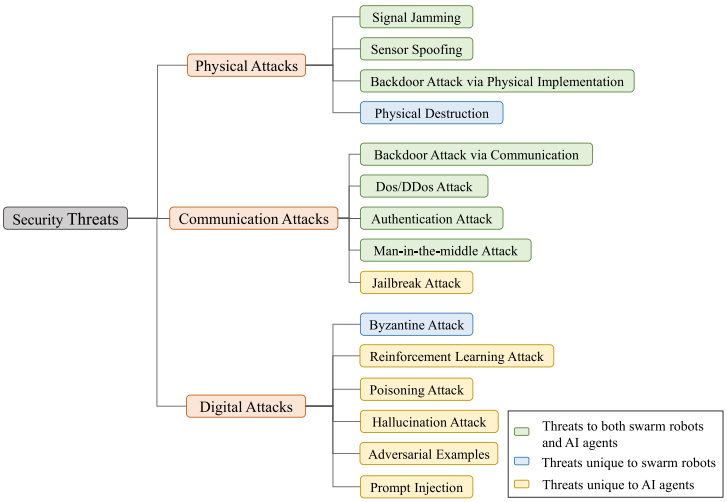

Figure 4. The security threats of swarm robots and AI agents.

or collaboration mechanisms, causing physical damage, failure or functional impairment.

(a) Common threats

The common threats include signal jamming attack, sensor spoofing attack, and backdoor attack via physical implementation.

i. **Signal jamming**: signal jamming attack is an attack method that introduces interference signals to disrupt normal signal transmission, thereby affecting the normal operation of sensors. For example, ground jammers can be used to interfere with drones through links, which can reduce the quality of drone signal reception or even cause drone communication failure [34]. For self-driving cars, adding air noise or ultrasonic noise to sensors can make detected objects disappear from the self-driving system, making it impossible to detect obstacles, resulting in parking or collision [35]. This attack assumes that the communication or sensing channel is either unencrypted or lacks robust noise filtering and authentication mechanisms. It is more feasible in open environments or cost-constrained deployments where physical access to signal space is possible and anti-jamming technologies are not deployed.

ii. **Sensor spoofing**: attackers can deceive the system into making wrong decisions by forging or tampering with sensor data. For drone systems, their global positioning systems can be maliciously deceived by false signals, and micro-electromechanical systems gyroscopes may be damaged by special frequency noise, thereby affecting their positioning and telemetry functions [36]. In addition, for visual sensors, real camera traces can be inserted into fake synthetic images based on generative adversarial networks (GANs) to deceive the camera detection model [37]. These attacks assume that the attackers are in close proximity to the target system and that affected sensors (e.g., GPS, gyroscopes, or cameras) are not equipped with sufficient signal authentication, filtering or cross-sensor validation mechanisms. For visual spoofing, it also presumes access to control of the visual input pipeline, such as compromised camera feeds or open digital input interfaces.

iii. **Backdoor attack via physical implementation**: physical backdoor attacks refer to the implantation of backdoor programs in hardware or physical devices, which allows attackers to gain control of the system and influence its behavior and decision-making process. Malicious manufacturers may deliberately leave backdoors in robot systems to quickly access robot systems in order to monitor robots and their owners without the knowledge of the owner [38]. For AI agents, backdoors are embedded into the model during the training of LLM, allowing attackers to manipulate model behavior using specific trigger patterns during inference [39]. This type of attack assumes a compromised supply chain or insider threat during the hardware manufacturing or model training phase. It is more feasible in scenarios where third-party vendors are involved without rigorous auditing or when models are trained using untrusted data sources or outsourced infrastructure.

(b) Threats unique to swarm robots

i. **Physical destruction**: physical damage attacks directly destroy the robot hardware or system, making it unable to continue to perform tasks, and undermining the collaborative capabilities of the swarm robot system. For example, important values stored in memory can be destroyed by directly heating the storage unit without any damage to the device [40]. Similarly, for rechargeable batteries installed inside the robot, potential battery exhaustion attacks can be performed to drain the battery energy, or legitimate batteries can be physically tampered with or swapped with incorrect batteries [41], directly causing damage to the robot system [42]. These attacks typically assume that the adversary has direct physical access to the robot, either during storage, maintenance, or idle phases, since live manipulation is difficult in mobile or secure deployment environments. Battery or memory tampering is more realistic

in warehouse, logistics, or deployment scenarios lacking environment monitoring or physical safeguards.

3.1.2. Communication attack. Communication attack refers to a type of security threat that interferes with, tampers with, forges or destroys the communication process between individuals, affecting their information transmission, coordination, or collaboration capabilities, thereby causing the entire system to fail, performance degradation, or mission failure.

(a) Common threats

The common threats include backdoor attack via communication, DoS/DDoS attack, authentication attack and man-in-the-middle attack.

i. **Backdoor attack via communication**: attackers can insert malicious code or commands through the communication network to control system behavior and affect information transmission. Internal vulnerabilities in the communication protocol can allow attackers to gain unauthorized access to the internal network of the robot and intercept or modify any transmitted data [43]. For example, in ABB RobotWare [44], various vulnerabilities found in its industrial network gateway proved the feasibility of a complete remote attack on the manufacturing robot system. Similarly, for AI agents, data containing different triggers can be transmitted to AI models to achieve the purpose of a backdoor attack [45]. This type of attack assumes that the communication protocol lacks end-to-end encryption or input validation, and that the system is exposed to an open or semi-trusted network environment (e.g., industrial IoT, public APIs, or edge devices). Attack feasibility increases significantly in systems that allow remote access without strict authentication or where protocol stacks are outdated or unpatched.
ii. **DoS/DDoS attack (denial of service/distributed denial of service)**: DoS/DDoS attacks refer to sending a large number of requests or data to the system, causing service paralysis or resource exhaustion, so that swarm robots or AI agents cannot continue to perform tasks. For robot systems that join the IoT, attackers can block the communication path by sending a large amount of data to some of the nodes; based on the attack method, DoS attacks in AI agents can be divided into data-oriented DoS attacks (such as Sponge Examples) and flooding-oriented DoS attacks (such as Intelligent Botnet), which destroy the availability of agent services by significantly increasing resource consumption [46]. These attacks generally assume that the robots or AI agents expose accessible interfaces (e.g., open ports, public IPs, or APIs) without rate-limiting, traffic shaping, or anomaly detection. They are especially effective in systems deployed over the public internet or loosely regulated internal networks where adversaries can generate high-volume or specifically crafted traffic without immediate detection or blocking.
iii. **Authentication attack**: attackers can forge identity information or tamper with communication data to bypass authentication mechanisms and potentially steal sensitive information or control systems. The design of some robot application interfaces does not require a login portal, allowing anyone to access them remotely [47]. In particular, for medical robots, malicious robots are introduced into the system by forging identities, thereby harming patients through surgery or *in vivo*, by providing incorrect medications, or by reporting medical data to unauthorized entities [48]. For AI agents, attackers can disguise themselves as legitimate users to gain control of the agent and then obtain legitimate user information or control the system to harm the user. These attacks assume weak or absent identity verification mechanisms, such as default or hardcoded credentials, lack of mutual authentication, or missing access control layers. It is particularly plausible in legacy systems, hospital intranets, or open-access deployments where internal trust is assumed and endpoint authentication is under-enforced.
iv. **Man-in-the-middle attack**: attackers insert themselves between the communicating parties, usually acting as a 'man-in-the-middle', and can listen to and possibly modify or intercept the communication content between the two without being noticed. In a swarm robot system, each robot usually relies on wireless communication for coordination and information exchange. If an attacker acts as a middleman, it will cause system data leakage or malfunction. In AI agent systems, especially in scenarios involving remote control, sensor data, command transmission, etc., a middleman attack may cause the agent's behavior to change, such as modifying the instructions of the autonomous driving system, causing the vehicle to make dangerous decisions and cause an accident. This type of attack assumes that communications are transmitted over unencrypted or poorly secured wireless channels, and that the attacker has physical proximity or network access to intercept traffic. It is especially plausible in Wi-Fi, BLE, or open radio frequency systems without mutual authentication, certificate pinning, or strong key exchange protocols.

(b) Unique to AI agents

i. **Jailbreak attack**: AI agents usually implement inherent predefined rule restrictions through model alignment technology to prevent the generation of harmful or malicious content [46]. Jailbreak attacks bypass the above security restrictions to cause the agent to generate harmful or malicious content. Studies have shown that for GPT-3.5 and GPT-4, two long-term jailbreak attacks can achieve a 99% attack success rate [49]. This attack assumes that the adversary has interactive access to the AI agent (e.g., via chatbot interface or API) and the ability to submit carefully crafted prompts over multiple turns. It is especially feasible in systems lacking robust prompt sanitization, session-level monitoring, or adversarial training against alignment circumvention techniques.
ii. **Tool and API misuse**: AI agents often invoke external tools (e.g., browsers, code runners, databases) to fulfill multi-step tasks [50]. If tool usage is not adequately authenticated, attackers can manipulate agent behavior through malicious API responses or system-level commands. This

attack assumes the agent lacks output validation or tool execution isolation. It is especially likely in tool-augmented agents that directly interface with OS-level utilities or financial systems.

3.1.3. Digital attack. A digital attack is an attack on digital systems or algorithms related to information processing, storage, transmission, and decision-making that interferes with or destroys a system's ability to perform tasks, make decisions, collaborate, or interact with other systems.

(a) Unique to swarm robots

i. **Byzantine attack**: the most common attack in swarm intelligence systems is the Byzantine attack [51–53]. In a Byzantine attack, an attacker can interfere with the normal operation of a distributed system through malicious behavior of nodes, such as the spread of contradictory messages and forged data. This attack manifests itself as inconsistency in the states of individual system nodes, failure of consensus, or the spread of false messages. There are three main types of Byzantine attacks. First, forged information attack, in which malicious nodes mislead other nodes to make wrong decisions by sending false state information or data. Second, communication tampering attack, in which malicious nodes selectively discard or refuse to forward key information, blocking system communication. Third, Sybil attack [54], in which malicious nodes create multiple false identities to interfere with the system's consensus process with majority weight. This class of attacks assumes that the swarm lacks a robust trust management or identity verification mechanism and that attackers can either compromise legitimate nodes or inject new ones into the network. Such conditions are more likely in open, ad hoc, or large-scale decentralized deployments where individual nodes cannot be fully authenticated.

(b) Unique to AI agents

i. **Adversarial examples**: attackers trick AI models into making incorrect predictions or decisions by carefully constructing input data (such as images, audio, sensor data, etc). Such attacks usually target deep learning models and can cause AI agents to make incorrect decisions that are difficult to detect under normal circumstances. LLM is able to perform timely semantic parsing on code, so it is easily affected by adversarial inputs, especially inputs generated by sentence-level perturbations [55]. This attack assumes the model receives inputs from untrusted sources and lacks sufficient adversarial training or input filtering. It is more likely in open APIs or interactive systems.

ii. **Poisoning attacks**: poisoning attacks can be divided into two categories: data poisoning and model poisoning. By injecting malicious data into the training data, the attacker disrupts the learning process of AI agents, thereby destroying the accuracy of the model and affecting its performance in the actual environment. For example, after injecting poisoned data into the database, using only one million of the poisoned samples can bring a 90% attack success rate [56]. In a distributed environment, the attacker can imitate the benign agent and upload the poisoned model update in each round of communication, resulting in the deterioration of the performance of the global agent [57]. This attack is especially feasible in federated or continual learning systems without authentication or anomaly detection, and it assumes weak data validation or unsecured model update processes.

iii. **Hallucination attack**: the hallucinations of AI agents can be divided into active hallucinations and passive hallucinations. Active hallucinations occur when the user inputs normally, but due to certain biases in the data and knowledge in the training process, or defects in the training and reasoning process itself, the agent deviates from the user input and generates erroneous or illogical outputs in the context of the real world [19]. Passive hallucinations occur when the attacker induces the system to generate false, unfounded or misleading output results by inputting specific prompts, thereby interfering with its judgment and decision-making. Passive hallucinations are especially feasible in open-ended systems that accept free-form user input without strong output filtering. These attacks assume the model lacks mechanisms for factual grounding or real-time validation.

iv. **Reinforcement learning attacks**: in a reinforcement learning system, AI agents interact with the environment, select actions based on the state, learn based on the reward signal, and adjust the policy to maximize the cumulative reward [19]. Attackers change the behavior of AI agents by interfering with the reward signal. For example, in an autonomous driving system, an attacker may encourage the system to choose dangerous driving behaviors by modifying the reward signal, such as driving fast or ignoring traffic signals for higher rewards. This attack assumes that the reward signal is externally observable or modifiable, and that the agent lacks verification mechanisms for reward integrity, and it is more likely in simulated or loosely coupled training environments.

v. **Prompt injection**: in natural language processing models (especially LLMs), the input prompts determine the output of the model. By injecting specific text or instructions, attackers can destroy the original intention or purpose of the model and force it to generate harmful or incorrect output. In addition to traditional injection attacks, Liu *et al* proposed Houyi [58], an innovative black-box rapid injection attack based on traditional Web injection techniques, which revealed serious attack consequences such as unlimited arbitrary use of LLMs and rapid theft. This attack assumes that the model processes untrusted or mixed-source input, such as user content combined with system instructions. It is particularly feasible in applications lacking prompt separation, user role isolation, or output validation.

3.2. Privacy threats

For swarm robotic systems, critical assets include sensory data streams, actuation and control signals, inter-agent

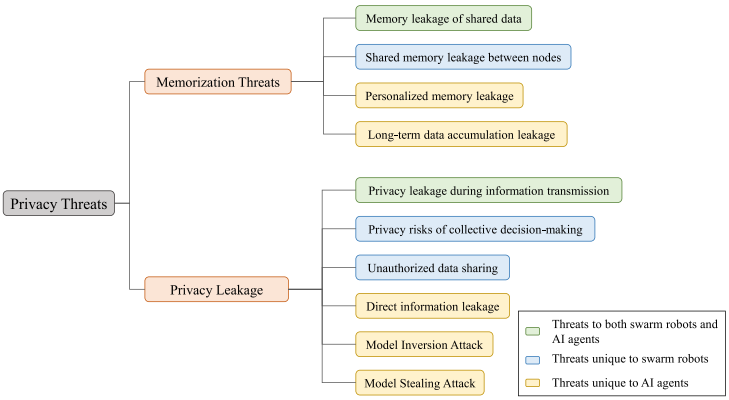

Figure 5. The privacy threats of swarm robots and AI agents.

communication channels, and the underlying coordination and consensus protocols. In the context of AI agents, key assets requiring protection comprise model parameters (both intermediate and final), proprietary or sensitive training datasets, inference outputs, and embedded decision-making algorithms or policies. However, memorization threats and privacy leakage represent significant vulnerabilities that may arise across the physical, communication, and digital layers of the system lifecycle. The primary categories of privacy threats are illustrated in figure 5.

3.2.1. Memorization threats. Both swarm robots and AI agents need to store and use certain data from users, the environment, and other parts of the system to perform tasks, so they will face privacy risks in shared data memory and information dissemination.

(a) Common threats

i. **Memory leakage of shared data**: for both swarm robots and AI agents, some user data needs to be stored and used. Although this memory function helps to improve the service quality, it also brings potential privacy risks. For example, the system may inadvertently remember the user's sensitive information, which may be leaked to unauthorized third parties if it is not cleared in time. In addition, the heterogeneous multimodal data collected and stored in the system increases the complexity of privacy protection, because the protection mechanism applicable to text may not be applicable to the protection of image [15]. This threat assumes that stored data is not encrypted, access control is weak, or memory persistence lacks automated clearing mechanisms. The risk is amplified in systems that continuously log or cache user interaction data across modalities.

(b) Threats unique to swarm robots

i. **Shared memory leakage between nodes**: for swarm robots, one of the unique challenges is shared memory and communication between nodes. Since such systems rely on collaboration between multiple nodes to complete tasks, improper data management between nodes or the lack of an effective coordination mechanism may lead to sensitive information leakage. This risk is particularly prominent when nodes frequently exchange data. In addition, if the system fails to clearly define the scope of data authorization, or if there is a lack of data protection between nodes, memory information may be leaked in situations where it should not be shared. This attack assumes the swarm system does not enforce strict access control, encryption, or communication isolation between nodes. It is especially feasible in dynamic or ad hoc swarm networks with limited central coordination.

(c) Threats unique to AI agents

i. **Personalized memory leakage**: memory interactions in AI agent systems involve storing and retrieving information throughout the agent's use, which involves three basic stages. 1) The agent collects information from the environment and stores it in memory; 2) Once stored, the agent processes this information to convert it into a more usable form; 3) The agent uses the processed information to inform and guide its next action. In other words, memory interactions allow AI agents to record user preferences, absorb valuable information from previous interactions, and use this acquired knowledge to improve service quality [59]. If these personalized memories are not effectively protected (for example, unencrypted or uncleaned), they may cause the user's private information to be inadvertently leaked. It is more likely in long-running personalized agents that retain state across sessions.

ii. **Long-term data accumulation leakage**: over time, AI agents will gradually accumulate a large amount of user data, forming a huge database of user private information. This data may be abused or leaked without adequate privacy protection measures. If long-term storage lacks proper data minimization, encryption, or audit mechanisms. The risk increases when user data is retained indefinitely for personalization or analytics without periodic review or anonymization.

3.2.2. Privacy leakage. Privacy leakage is a significant concern for both swarm robots and AI agents, particularly in how sensitive data is shared, transmitted, and stored.

(a) Common threats

i. **Privacy leakage during information transmission**: both share information between different nodes or agents. Once the information is not properly encrypted during transmission, or there are loopholes in the transmission mechanism itself, malicious attackers will have an opportunity to take advantage of it.

(b) Threats unique to swarm robots

i. **Privacy risks of collective decision-making**: swarm robot systems usually achieve mission goals through collective decision-making or consensus algorithms. If these decisions rely on sensitive user data and the algorithm design is not secure enough, information in the decision-making process may be inadvertently exposed. To reach a consensus, robots must exchange their status information with each other on public communication channels. Adversaries can monitor public communication channels and obtain private information of individual robots, and other nodes may also steal information [60].

ii. **Unauthorized data sharing**: for swarm robot systems where robots with sensitive identities exist in the group, attackers can observe and collect the robot's motion data and infer sensitive identities such as the leader, thereby obtaining sensitive information about the system's identity [61].

(c) Threats unique to AI agents

i. **Direct information leakage**: AI agents collect a lot of personal information from users, such as health, location, shopping history, etc. If this information is not encrypted or effectively protected, malicious attackers or unauthorized third parties may obtain this sensitive data, resulting in privacy leakage.

ii. **Model inversion/Model stealing attack**: AI agents optimize their decision-making or behavior models based on this collected information and historical data. If these models store sensitive data of users and the models are not adequately protected (such as encryption or isolation), attackers may obtain user privacy through model stealing attacks, model inversion attacks, and other means. For example, an attacker can extract model information, by querying the model and observing the corresponding responses, and then steal the target model without accessing the original data [62]; for malicious users with access to vector databases and text insertion pairs in the model used to generate data, malicious users can learn a function that copies text from the embedding, thereby obtaining user privacy [63].

4. Countermeasures of swarm robots and AI agents

Implementing effective security strategies and privacy-preserving techniques is crucial for swarm robotics and AI agents. This section explores these strategies and techniques, along with their integrations.

4.1. Security strategies and privacy-preserving techniques

While there is a degree of overlap between security strategies and privacy-preserving techniques, their objectives and methodologies differ:

- Security strategies primarily focus on protecting systems from external threats, unauthorized access, and cyber-attacks.
- Privacy-preserving techniques are designed to ensure data confidentiality and minimize the risk of sensitive information exposure.

Recent research comprehensively demonstrates the definition, scope, and strategies of digital security from a design perspective [64], while other work provides a literature review of cybersecurity in the domains of AI, blockchain, and cloud computing [65]. Inspired by these studies, our paper presents a detailed comparison of security strategies and privacy-preserving techniques, along with their relevance to swarm robots and AI agents, as shown in table 3. To provide a comprehensive classification, we categorize these methods based on their objective, as well as their implementation across the physical layer, communication layer, and application layer, including specific use cases. This classification provides a structured framework for evaluating security and privacy techniques in swarm robotic systems and AI agents, ensuring both resilience against attacks and compliance with privacy constraints in real-world applications.

4.2. Countermeasures to swarm robots

4.2.1. Security threat countermeasures. The security of swarm intelligence systems is critical to ensuring their stable operation and gaining widespread trust and adoption in practical applications. Since a single security mechanism can often defend against multiple types of attacks, we categorize security solutions based on their underlying defense techniques in this section, as shown in table 4.

- **Neighborhood filtering**

Achieving consensus in swarm systems is crucial for coordinated task execution. Neighborhood filtering techniques are widely adopted to identify and isolate malicious agents by comparing a node's behavior with that of its neighbors.

For example, the **mean subsequence reduced (MSR)** algorithm [66] effectively filters out anomalous values from potentially compromised neighbors. Building on this, **joint robustness** [67] extends the static graph analysis to dynamic topologies, improving resilience in time-varying communication networks. The **weighted MSR algorithm** [68] further strengthens detection by incorporating physical signal fingerprints; however, it assumes prior signal profiles and is limited in environments with restricted information propagation.

Table 3. Comparison of security strategies and privacy-preserving techniques for swarm robots and AI agents.

Category	Security strategies (protecting system & network)	Privacy-preserving techniques (protect data & identity)
Objective	Prevent unauthorized access, cyber-attacks, and system vulnerabilities	Ensure data confidentiality, minimize exposure, and prevent re-identification
Physical layer (Hardware, sensors, actuators)	- Secure boot & firewalls - Physical access control (e.g., biometric authentication, smart locks, secure element) - Trusted execution environments - Power side-channel attack mitigation	- Anonymous authentication (e.g., zero-knowledge proofs, unlinkable credentials)
Communication layer (Network, data transmission, swarm coordination)	- Byzantine Fault Tolerance - Blockchain - Secure communication protocols (e.g., TLS, VPN, IPsec) - End-to-end encryption (AES, RSA)	- HE - Onion routing (e.g., Tor) - Private set intersection - Federated learning
Application layer (AI models, swarm intelligence, data processing)	- Access control & authentication (Threshold limitation, multi-factor authentication, RBAC) - Input/output filtering - Data sanitization - Adversarial training - Reinforcement learning	- MPC - DP - ZKPs - Federated learning - Poisoned samples filtering - Knowledge distillation - Model watermarking
Use cases in swarm robots	Secure communication between agents, encrypted control signals, fault tolerance in decentralized networks	Privacy-aware mission planning, secure sharing of sensor data without exposing sensitive locations
Use cases in AI agents	Protection against adversarial attacks, encrypted model updates, secure API access	Privacy-preserving AI training, DP in data processing

To address these limitations, a **decentralized blacklist protocol** [69] enables robots collaboratively share accusation data and execute a graph-matching algorithm to generate blacklists. Similarly [70], introduces two distributed schemes based on a **two-hop** communication mechanism, enabling nodes to detect abnormal behavior in their neighbors and achieve resilient consensus. These approaches have been effectively implemented in swarm robots and UVA cooperation. However, scalability in large networks and adaptability to highly dynamic conditions remain open challenges.

- **Physical signal analysis**

Physical signals and infrastructures are effective for detecting malicious nodes, particularly in the case of Sybil attacks [71] proposes a defense mechanism leveraging **wireless signal analysis**. This approach extracts unique **spatial fingerprints** from the scattering and absorption characteristics of wireless signals in the environment, making them difficult for attackers to manipulate, thereby enhancing robustness and resistance to tampering. In real-world robotic swarms, ScatterID [72] demonstrates a lightweight system solution using battery-free **backscatter tags** for single-antenna robots to enable secure identification even in resource-constrained systems.

In crowdsourced intelligent transportation systems [73], explores the use of noise data from traditional sensor infrastructures and vehicle dynamics inferred from crowdsourced data to detect and counteract virtual congestion and path-planning disruptions caused by malicious actors reporting fake 'ghost' vehicles. While effective, these approaches often require hardware integration and may struggle in environments with high signal variability or interference.

- **Blockchain**

Blockchain technology has been increasingly applied to enhance the security of swarm intelligence systems. The first integration of swarm intelligence and blockchain was proposed by Castelló Ferrer [74], who highlighted blockchain as a key technology for advancing swarm robotics, particularly in secure communication, distributed decision-making, and innovative business models. Subsequently [75], provides the first **real-world proof-of-concept** for using blockchain in robot coordination, detailing its implementation and experimental validation.

Furthermore [76], introduces a **blockchain- and token-based security** framework, where smart contracts manage token allocation among robotic agents, rewarding honest participants and eliminating Byzantine nodes. Similarly [77],

Table 4. Summary of key literature on security threat countermeasures to swarm robots.

Countermeasure	Publication	⋆ Method • Advantages ◦ Limitations
Neighborhood filtering	[66] MSR algorithm	⋆ MSR algorithm filters out abnormal values from neighbors. • Robust against random failures and faulty nodes. ◦ Requires predefined threshold, may not adapt well to dynamic adversaries.
	[67] Joint robustness	⋆ Extends robustness concept to time-varying communication graphs. • Ensures resilience in dynamic networks. ◦ Computational overhead increases with graph complexity.
	[68] W-MSR algorithm	⋆ Incorporates physical signal fingerprint analysis into MSR. • Enhances security by leveraging physical-layer properties. ◦ Limited scalability; depends on stable physical signals.
	[69] Decentralized blacklist protocol	⋆ Uses graph-matching algorithms and shared accusation data. • Effectively isolates malicious agents in collaborative environments. ◦ Potential false positives in noisy conditions.
	[70] Two-hop communication mechanism	⋆ Leverages two-hop neighbor data to detect malicious nodes. • Reduces direct influence of malicious nodes in small networks. ◦ Additional communication overhead in large-scale systems.
Physical signal analysis	[71] Wireless signal-based detection	⋆ Uses wireless signal propagation characteristics for identity verification. • Hard to forge physical-layer features. ◦ Requires specialized hardware for signal analysis.
	[72] ScatterID lightweight system	⋆ Uses ultra-lightweight backscatter tags on robots. • Energy-efficient and lightweight for mobile agents. ◦ Limited deployment in environments without signal interference.
	[73] Noise data usage	⋆ Uses sensor noise data and crowdsourced vehicle dynamics to verify congestion reports. • Effectively filters out fake traffic congestion attacks. ◦ Requires high participation for reliable detection.
Blockchain	[74] Basic blockchain	⋆ Proposes blockchain-based secure communication and decision-making. • Ensures data integrity and prevents tampering. ◦ High computational and storage costs.
	[75] Proof-of-concept	⋆ Proof-of-concept study on robot coordination using blockchain. • Provides decentralized trust mechanism. ◦ Latency issues in real-time applications.
	[76] Smart contract	⋆ Uses smart contracts to manage token distribution. • Encourages cooperative behavior through incentives. ◦ Token economy may be vulnerable to manipulation.
	[77] Meta-controller	⋆ Implements smart contracts to prevent identity forgery. • Prevents identity spoofing in multi-agent systems. ◦ Smart contract vulnerabilities can still be exploited.
	[78] High-throughput protocol	⋆ Develops a blockchain-based communication framework for mobile networks. • Enables secure decentralized communication. ◦ Scalability issues for large swarm sizes.
Reinforcement learning	[79] Two-stage intrusion detection	⋆ Combines signature-based and anomaly-based intrusion detection using deep learning. • Provides adaptive detection of novel attacks. ◦ Requires continuous retraining with new attack patterns.
	[80] Adversarial deep reinforcement learning	⋆ Uses LSTM for attack prediction and GANs to model threats. • Enhances resilience against evolving adversarial strategies. ◦ High computational requirements; adversarial attacks on the model itself are possible.
	[81] Reinforcement learning for UAV security	⋆ Enables UAVs to autonomously learn and respond to threats. • Adaptive security mechanism for UAVs. ◦ Requires extensive training and real-world validation.

(Continued.)

Table 4. (Continued.)

Countermeasure	Publication	⋆ Method • Advantages ◦ Limitations
Security protocols	[82] Coordinators and state controllers	⋆ Uses a decentralized architecture without a central trust server. • Eliminates reliance on a single point of failure. ◦ Requires high system complexity to maintain security.
	[83] QR-swarm protocol	⋆ Implements a distributed authentication scheme based on Fiat-Shamir. • Provides a lightweight identity verification mechanism. ◦ Vulnerable to precomputed attacks if key updates are infrequent.
	[84] Unif-swarm protocol	⋆ Extends QR-Swarm to provide a unified zero-knowledge proof framework. • Improves privacy while maintaining authentication. ◦ Computational overhead compared to non-cryptographic methods.
	[85] DL-swarm & CDH-swarm protocols	⋆ DL-Swarm extends Unif-Swarm; CDH-Swarm leverages Computational Diffi–Hellman. • Enhances authentication security in swarm networks. ◦ Requires significant cryptographic processing power.

employs **smart contracts as meta-controllers** in collective perception scenarios to prevent identity forgery in robotic swarms. In another study [78], develops a high-throughput communication framework for decentralized mobile ad hoc networks, using blockchain as a security foundation for swarm robotic systems composed of Pi-puck robots. However, challenges remain in balancing latency, energy consumption, and throughput, particularly in time-critical applications such as UAV swarms or rescue robots.

- **Reinforcement learning**

Reinforcement learning has also been explored for detecting and mitigating attacks in swarm intelligence systems. For instance [79], proposes a **two-stage intrusion detection system** consisting of a signature-based detection module and an anomaly detection module. The latter leverages deep neural networks to detect command deviations from expected behavior. Additionally [80], develops an adversarial deep reinforcement learning algorithm to improve resilience against malicious interventions. Specifically, each robot employs long short-term memory (LSTM) networks to predict distance variations caused by external interference and GANs to simulate and assess potential attacks, ensuring the swarm system remains stable in adversarial environments. Furthermore [81], implements a reinforcement learning-based secure UAV system, allowing drones to autonomously learn and respond to the behaviors of both targets and potential intruders.

- **Security protocols**

In the domain of secure protocol design for swarm intelligence systems [82], introduces a security architecture that eliminates the need for a central trust server. To enable cooperative behavior among multiple robots while ensuring non-repudiation, traceability, and resistance to malicious attacks, the study proposes using **coordinators and state controllers** to record the state changes of participating robots in **Winternitz Stack,** providing historical records and verifiable logs.

For node security verification, researchers have proposed multiple security protocols, including **QR-Swarm** [83], **Unif-Swarm** [84], **DL-Swarm**, and CDH-Swarm [85]. QR-Swarm is a distributed authentication protocol based on the Fiat-Shamir framework, while Unif-Swarm extends QR-Swarm into a unified zero-knowledge proof protocol, offering a more comprehensive security framework applicable to various swarm intelligence systems. DL-Swarm is a specific implementation of Unif-Swarm, whereas CDH-Swarm further enhances protocol security certainty by leveraging computational Diffie–Hellman security mechanisms. These protocols are theoretically sound and show strong potential in controlled testbeds. However, their integration into heterogeneous swarms with limited bandwidth and energy constraints needs further validation through field trials.

4.2.2. Privacy threat countermeasures. Data leakage in swarm intelligence systems primarily occurs during the consensus formation process. Existing consensus algorithms require each node in the system to exchange explicit state information with its neighbors, leading to potential privacy breaches. To address this issue, various privacy-preserving methods have been proposed for consensus algorithms, including HE, DP, observability-based protection, and federated learning. The following sections elaborate on these approaches and the summary is shown in table 5.

- **Homomorphic encryption**

HE, a widely used end-to-end encryption method, plays a crucial role in data transmission and consensus computation. Study [86] proposed a method for undirected networks that employs **partial HE** to enable secure interactions between nodes in a system without relying on an aggregator, thereby preventing privacy leaks. Similarly, by integrating a dynamically changing quantizer with the Paillier cryptosystem, study [87] introduced an **encrypted control algorithm** for solving the average consensus problem in distributed systems with strongly connected directed graphs as their communication topology. This approach not only effectively addresses the

Table 5. Summary of key literature on privacy threat countermeasures to swarm robots.

Countermeasure	Publication	⋆ Method • Advantages ◦ Limitations
Homomorphic encryption	[86] Secure P2P consensus	⋆ Partial HE for secure P2P interactions. • Enhances decentralization and privacy. ◦ Computational and communication overhead.
	[87] Encrypted average consensus	⋆ Paillier cryptosystem with dynamic quantizers. • Accurate consensus with lower complexity. ◦ Increased encryption overhead.
	[88] Logistics privacy optimization	⋆ Local HE with swarm intelligence for logistics • Optimizes while preserving privacy. ◦ High computational burden.
Differential privacy	[89] Event-triggered DP consensus	⋆ DP-enhanced consensus with event-triggered updates. • Reduces communication and computation cost. ◦ Accuracy loss due to DP noise.
	[90] LDED encoding for DP	⋆ Logarithmic encoding-decoding to reduce DP errors. • Balances privacy and accuracy. ◦ Minor quantization errors.
	[91] DP for swarm intelligence	⋆ A general framework of DP on swarm intelligence. • Maintains performance while preserving privacy. ◦ Efficiency trade-offs.
Observability-based protection	[92] Privacy index	⋆ Quantifies node observability for privacy analysis. • Provides a quantitative metric for privacy assessment. ◦ Increased node cooperation can lead to higher communication costs
	[93] Opinion-inspired privacy	⋆ Iterative consensus with opinion-based privacy • Privacy without encryption overhead. ◦ Less effective in highly dynamic or large-scale systems.
Federated learning	[94] FL + Swarm optimization	⋆ FL with swarm optimization for distributed tasks. • Enables privacy-preserving distributed optimization. ◦ Potential data and model heterogeneity challenges.
	[95] P2P asynchronous FL for robots	⋆ Reputation-aware coordination for securing intelligent mobile robots in 5 G networks. • Secure, reputation-aware dynamic coordination. ◦ Vulnerable to model poisoning and inference attacks.
	[96] Defending FL against poisoning	⋆ Knowledge distillation and feature map filtering to mitigate feature map poisoning attacks. • Enhances model robustness against poisoning attacks. ◦ Does not fully address data heterogeneity issues.
	[97] Decentralized FL with Reputation	⋆ Using unlinkability and reciprocity principles, with a decentralized reputation management system. • Reduces computational cost and detects malicious updates. ◦ May require additional incentives to ensure node compliance.

average consensus problem but also reduces computational complexity while maintaining accuracy. Furthermore, study [88] combined a **bottom-up** swarm intelligence approach with **local rule-based** homomorphic encryption techniques to achieve secure multi-party optimization in the logistics industry, enhancing both optimization efficiency and data privacy protection. Compared to traditional unencrypted optimization methods, this approach ensures the feasibility and effectiveness of optimization while maintaining data security. However, these encryption-based algorithms introduce additional computational and communication overhead.

• **Differential privacy**

Various studies have explored differential privacy techniques to protect user data privacy during the consensus formation process. For example, study [89] proposed a distributed **event-triggered** mechanism combined with a differential privacy consensus algorithm, effectively reducing the frequency of real-time communications and controller updates while safeguarding data privacy. This approach significantly reduces system communication and computational overhead compared to traditional periodic communication mechanisms. However, it may introduce accuracy issues. To mitigate the quantization errors caused by DP during data transmission and balance

privacy protection with consensus accuracy, study [90] proposed a **logarithmic dynamic encoding-decoding** scheme.

Beyond consensus formation, differential privacy techniques have also been applied to optimizing swarm intelligence algorithms. Study [91] was the first to integrate DP into swarm intelligence, introducing a general framework for **differentially private swarm intelligence algorithms**. This framework allows for individual data privacy protection while maintaining optimization performance during the optimization process. While DP is computationally lightweight and scalable, its effectiveness depends heavily on the privacy budget and system noise tolerance. High noise levels can impair coordination, limiting applicability in high-precision tasks.

- **Observability-based protection**

Observability-based privacy protection methods leverage the observability theory of dynamic systems to study how state information leaks in system dynamics, preventing adversaries from inferring node privacy information through system evolution observations. Study [92] proposed a 'privacy index' to quantify the minimum number of agents needed to reconstruct a system's initial state. A higher privacy index implies greater resistance to inference attacks, but may increase the system's communication and computational complexity. A practical application of this concept is seen in collaborative drone surveillance missions, where preserving the location of high-value targets is crucial. Ensuring that no single drone's data stream can reveal mission-critical details without aggregating data from several nodes can deter adversarial inference.

Additionally, study [93], inspired by the public and private opinion framework in social networks, proposed a novel **iterative** algorithm. By designing separate interaction variables and actual state variables, the algorithm achieves privacy protection naturally without requiring additional privacy-preserving technologies, enhancing the effectiveness of privacy protection and explainability of framework. These methods are lightweight, but assume adversaries possess observational capabilities that may not always be practical in constrained environments.

- **Federated learning**

Federated learning, an effective solution to the data silo problem, also plays a significant role in protecting data privacy in swarm intelligence systems. The core idea of federated learning is to enable collaborative model training among distributed clients without sharing raw data. For example, study [94] combined particle swarm optimization with federated learning to develop a privacy-preserving swarm intelligence optimization algorithm.

Similarly, to enhance the security and robustness of intelligent mobile robots in 5 G and future networks, study [95] introduced a peer-to-peer (P2P) privacy-perceiving asynchronous federated learning framework. This framework utilizes a reputation-aware coordination mechanism to dynamically organize multiple intelligent devices into a virtual swarm intelligence system, ensuring an encrypted P2P federated learning process. However, traditional federated learning frameworks still face security threats such as poisoning attacks, model inversion attacks, gradient leakage, and inference attacks.

To counter these threats, researchers have integrated DP, secure MPC, and HE. For instance, researchers proposed a feature map poisoning attack and a dual defense mechanism against federated prototype learning (FedProto) in study [96]. Their approach explored the vulnerability of FedProto to feature map poisoning attacks and improved the prediction accuracy of compromised models by 1–5 times using **full-knowledge distillation** and **feature map filtering**. However, this solution does not account for challenges related to data and model heterogeneity.

Similarly, study [97] proposed a decentralized federated learning framework that incorporates unlinkability and reciprocity principles to ensure privacy and security. This framework utilizes a **decentralized reputation management** system to incentivize compliance among nodes, maintaining model integrity while protecting privacy. By detecting malicious updates and reducing computational costs, it outperforms DP and HE.

4.3. Countermeasures to AI agents

4.3.1. Security threat countermeasures. Due to the diversity of attacks, security defense methods also vary significantly. We primarily categorize defense methods into filtering, DP, fine-tuned reinforcement learning, and others, as shown in table 6.

- **Filtering**

Filtering is an effective defense mechanism against various attacks, including DoS attacks, backdoor attacks, data poisoning attacks, and hallucinations. The core idea of filtering is to detect and eliminate **abnormal inputs and outputs**, ensuring that only legitimate data is processed by the model. This approach often incorporates **manual control over datasets** to enhance model reliability and security.

Several studies have explored filtering as a countermeasure. Zeng *et al* [99] propose **AutoDefense**, a **multi-agent framework** designed to defend against jailbreak attacks by filtering harmful LLM responses without modifying user inputs, ensuring robust content moderation without restricting user intent. Additionally, **Neural Cleanse** [100] enhances model interpretability and defense against backdoor attacks by identifying *anomalous activations* in neurons. Its reverse-engineering approach detects suspicious triggers with minimal false positives, though its scalability to large LLMs remains limited.

Additionally, **SelfCheckGPT** [101] is introduced as a self-verification method to detect hallucinations by comparing consistency across multiple generated responses. This method enables real-time identification of factual inconsistencies, improving the trustworthiness of AI-generated content. Filtering offers high interpretability and task-specific protection. However, it requires domain knowledge for

Table 6. Summary of key literature on security threat countermeasures to AI agents.

Countermeasure	Publication	⋆ Method • Advantages ◦ Limitations
Filtering	[99] AutoDefense	⋆ Multi-agent framework filters harmful LLM responses without altering user inputs. • Maintains user intent while ensuring robust content moderation. ◦ May not fully detect adversarially crafted harmful inputs.
	[100] Neural Cleanse	⋆ Identifies and neutralizes compromised neurons sensitive to backdoor triggers. • Effectively mitigates backdoor attacks and enhances model integrity. ◦ Computationally expensive and may not generalize to all backdoors.
	[101] SelfCheckGPT	⋆ Self-verification by cross-checking multiple responses for consistency. • Improves factual reliability of AI-generated content. ◦ Limited to detecting inconsistencies rather than preventing hallucinations.
Differential privacy	[102] DP-gradient smoothing	⋆ Adds DP noise to training data or gradients to reduce the impact of poisoning. • Strengthens model robustness against data poisoning and backdoor attacks. ◦ Can degrade model accuracy due to added noise.
Fine-tuned reinforcement learning	[103] Three-round fine-tuning	⋆ Three-round fine-tuning to minimize harmful responses and enhance performance. • Balances security with improved model responses. ◦ Requires large-scale human feedback data.
	[104] Code repair RL	⋆ Uses semantic and syntactic reward mechanisms for secure program repair. • Enhances robustness and correctness of generated code. ◦ Computational overhead in RL optimization.
	[105] Offline RL alignment	⋆ Uses FA, RWR, and CA to align models. • Aligns models with human preferences without direct interaction. ◦ Requires well-curated human feedback for effective alignment.
Robust optimization	[106] Robust instruction tuning	⋆ Fine-tunes multi-modal models using robust instruction tuning to mitigate hallucinations. • Enhances model reliability and reduces incorrect outputs. ◦ May require significant domain-specific data for tuning.
Blockchain	[107] Federated blockchain	⋆ Integrate Hyperledger Fabric and low rank adaptation hyperparameters. • Improve the security, transparency, and verifiability of the unlearning process. ◦ Computationally expensive and requires more settings.
Audit & red teaming	[108] ARCA	⋆ Uses discrete optimization techniques to systematically audit LLM behavior. • Identifies vulnerabilities and improves security through automated audits. ◦ Computationally expensive and may struggle with dynamic adversarial strategies.
Formal security policies	[109] Formal security guarantees	⋆ Introduce novel and flexible domain-specific languages for specifying security policies, thereby helping to prevent security vulnerabilities in real-world deployments. • A new domain specific language that allows flexible specification for security rules for agents. ◦ Different tasks require tailored security rules, which limit the generalizability of such formal security approaches.

rule-setting, and real-time filtering can be computationally intensive.

- **Differential privacy**

DP is an effective method for countering poisoning attacks. For example, adding DP noise to training data or gradients during the training process can enhance a model's robustness against both data poisoning and backdoor attacks. Xu *et al* [102] introduce a differentially private training method that smooths training gradients in text classification tasks. This serves as a general defense mechanism against data poisoning attacks, reducing the impact of maliciously manipulated data. However, DP is more commonly used for privacy preservation, ensuring that sensitive user data remains protected during model training.

- **Fine-tuned reinforcement learning**

Reinforcement learning with human feedback helps prevent hallucinations in AI agent systems by incorporating **external feedback and reward** mechanisms to regulate the behavior. Dai *et al* [103] propose a method designed to reduce harmful responses while enhancing model performance through a **three-round** fine-tuning process. Similarly, for security hardening and code robustness, study [104] introduces a reinforcement learning-based method for program-specific repair. This approach integrates **semantic and syntactic reward** mechanisms, focusing on both functional correctness and security enhancements in generated code.

Furthermore, Hu *et al* [105] present an offline learning framework based on human feedback, enabling LLM alignment without direct interaction with environments. This framework explores techniques such as filtering alignment (FA), reward-weighted regression (RWR), and conditional alignment (CA) to better align models with human preferences. There are other similar studies on aligning AI agents through reinforcement learning-based fine-tuning, such as [110, 111].

- **Others**

There are several other widely used countermeasures for enhancing the security and reliability of agent AI systems, such as robust instruction tuning, blockchain, audit and red teaming. Robust enhances model resilience by refining response patterns through curated prompts, reducing susceptibility to adversarial manipulation and misalignment [106]. A comprehensive survey on the use of blockchain for LLM security and privacy has been conducted [112]. Specifically, a federated trustchain, blockchain-enhanced LLM training, and an unlearning method have been proposed to enhance the security of AI agents [107]. Automated auditing and red teaming employ adversarial testing to identify vulnerabilities, reinforcing defenses against jailbreaks, prompt injections, and unintended outputs [108]. Formal security guarantees offer an alternative approach to building secure and controllable AI agents by introducing novel and flexible domain-specific languages for specifying security policies, thereby helping to prevent security vulnerabilities in real-world deployments [109]. It is a new domain specific language that allows flexible specification for security rules for agents. However, different tasks may require tailored security rules, which limit the generalizability of such formal security approaches.

4.3.2. Privacy threat countermeasures. The privacy threat countermeasures can be divided into different privacy, data anonymization, federated learning, and others, as shown in table 7.

- **Differential privacy**

DP provides a rigorous framework that introduces noise during the training or fine-tuning of LLMs, making it infeasible to extract the original training data. Several studies have explored DP-based fine-tuning for privacy preservation, such as **EW-Tune** [113], which optimizes noise injection. Given a finite number of compositions, EW-Tune induces less noise in stochastic gradient descent (SGD) compared to state-of-the-art methods, ensuring better utility while maintaining privacy.

To enhance privacy in LLM inference, Mai *et al* [114] propose the **Split and Denoise** method, leveraging local DP. SnD allows clients to introduce noise before transmitting embeddings to the server, which then returns denoised output embeddings for downstream tasks. However, most LLM privacy evaluations treat individual text records as the privacy unit, leading to inconsistent privacy guarantees when user contributions vary.

To ensure uniform privacy protection across users [115], investigates **user-level DP** using Group Privacy and User-wise DP-SGD schemes. Similar approaches employ two DP-SGD variants: example-level sampling with per-example gradient clipping and user-level sampling with per-user gradient clipping. These techniques enhance privacy protection by aligning guarantees at the user level rather than the record level.

- **Data anonymization**

Chen *et al* [116] introduce the HaS framework, a lightweight solution for prompt privacy protection. 'H(ide)' and 'S(eek)' represent its two core processes: hiding private entities for anonymization and seeking private entities for de-anonymization. To minimize the exposure of sensitive PII in prompts, **LegalGuardian** [117] employs named entity recognition (NER) techniques to mask and unmask confidential PII within text. Additionally, several other approaches enhance privacy in human-LLM interactions, including **IncogniText** [118], which implements privacy-enhancing conditional text anonymization through LLM-based private attribute randomization and **Adanonymizer** [119], designed to navigate and balance the trade-off between privacy protection and output quality in human-LLM interactions.

- **Federated learning**

Fine-tuning requires user behavior data, which poses significant privacy risks due to the incorporation of sensitive information. The unintended disclosure of such data could violate data protection laws and raise ethical concerns. Integrating federated learning with foundation models presents a promising

Table 7. Summary of key literature on privacy threat countermeasures to AI agents.

Countermeasure	Publication	⋆ Method • Advantages ◦ Limitations
Differential privacy	[113] EW-Tune	⋆ Optimizes noise injection in DP-based fine-tuning to reduce noise in SGD while preserving privacy. • Enhances utility while maintaining privacy. ◦ Limited to finite compositions, which may impact model accuracy.
	[114] SnD	⋆ Local DP-based method that introduces noise to embeddings before transmission and denoises outputs. • Enhances privacy in LLM inference while retaining usability. ◦ Privacy guarantees vary due to inconsistent record-level evaluations.
	[115] User-level DP	⋆ Implements Group Privacy and DP-SGD schemes using ELS and ULS for better privacy protection. • Provides uniform privacy guarantees across users. ◦ Potential trade-offs between privacy strength and model performance.
Data anonymization	[116] HaS framework	⋆ Uses "Hide" (anonymization) and "Seek" (de-anonymization) for prompt privacy protection. • Lightweight and efficient anonymization for prompts. ◦ Effectiveness depends on the robustness of entity detection.
	[117] LegalGuardian	⋆ Uses NER to mask and unmask PII within prompts. • Reduces PII exposure without degrading text quality. ◦ Accuracy depends on NER performance.
	[118] IncogniText	⋆ Implements LLM-based private attribute randomization for text anonymization. • Enhances privacy without significantly impacting readability. ◦ Potential loss of contextual information in anonymized text.
	[119] Adanonymizer	⋆ Balances privacy and output quality in human-LLM interactions. • Provides customizable privacy settings for different scenarios. ◦ Trade-off between privacy strength and model effectiveness.
Federated learning	[122] OpenFedLLM	⋆ Uses federated instruction tuning and value alignment with FL algorithms for privacy-preserving fine-tuning. • Enables decentralized fine-tuning without exposing raw user data. ◦ Higher computational overhead and synchronization complexity.
	[123] FL-GLM	⋆ Keeps input/output blocks on local clients and uses encryption to prevent embedding gradient attacks. • Protects against server-side attacks and data leakage. ◦ Encryption overhead may affect system efficiency.
Multi-party computation	[126] MARILL	⋆ Uses a minimized multi-party computation approach for secure LLM inference. • Reduces computational complexity in secure inference. ◦ MPC overhead can still be significant for large-scale models.
Watermark	[127] Double-i	⋆ Embeds watermarking during fine-tuning to protect model copyrights. • Enhances model ownership protection and integrity. ◦ May affect fine-tuning flexibility and model adaptation.
Communication protocol	[128] Agent2Agent (A2A) protocol	⋆ Provides robust communication. • Enhance secure interoperability among AI agents. ◦ Another layer of implementation.

solution to mitigate privacy leakage, and several surveys have comprehensively explored this approach [120, 121].

Specifically, Ye *et al* [122] introduced **OpenFedLLM**, which encompasses federated instruction tuning to enhance instruction-following capabilities, federated value alignment to align with human values, and seven representative FL algorithms. Similarly, study [123] proposes the **FL-GLM** method, which places the input and output blocks on local clients to prevent embedding gradient attacks from the server. Additionally, it employs key encryption during client-server communication to safeguard against reverse engineering attacks from peer clients.

Federated learning offers an effective approach to improving fairness. For example, FairFed [124]introduces a fairness-aware aggregation method designed to ensure equitable model performance across different sensitive groups. Similarly, the Fairness-aware Federated Clustering Algorithm (Fair-FCA) clusters clients in a way that enables a tunable trade-off between fairness and accuracy [125].

- **Others**

There are several other techniques for enhancing privacy, including blockchain [103], MPC, and watermarking. Rathee *et al* [126] propose the **MARILL** framework, a minimized multi-party computation approach for secure and private LLM inference. Various watermarking techniques for AI models have also been explored, as highlighted in the survey [129]. For example, the **double-i** watermark method has been proposed to protect model copyright during LLM fine-tuning [127].

Establishing secure communication channels between AI agents is crucial for enabling real-time interaction with external data sources and tools, while safeguarding against model vulnerabilities and data leakage. Recent studies have introduced a comprehensive, multi-layered security framework tailored to the Model Context Protocol—a standardized interface for AI systems to connect with external resources. This framework integrates defense-in-depth strategies, Zero Trust principles, stringent tool vetting, continuous system monitoring, and rigorous input/output validation to ensure robust protection throughout the AI lifecycle [98]. Similarly, researchers have proposed building secure agentical AI applications through robust communication protocols such as Google's Agent2Agent (A2A), which enhances secure interoperability among AI agents.

4.4. Comparison and cross-domain learning

4.4.1. Comparison of security design. The security design principles differ due to the distinct architectural and operational characteristics of swarm robots and AI agents.

Swarm robotic systems are designed with low computational power and operate in real-time, dynamic environments, often facing bandwidth limitations and hostile conditions. Security countermeasures in swarm robotics emphasize **secure communication, redundancy, distributed decision-making**, and **fault tolerance**. These systems deploy **multi-agent consensus** to ensure reliability, even in the presence of faulty or malicious agents. **Blockchain technology** is often utilized for decentralized, tamper-resistant communication and verification, ensuring transparency and trust in data exchanges between robots. **Stigmergic coordination**, where actions are coordinated through environmental cues, allows swarm robots to detect and respond to anomalies passively, contributing to self-healing and adaptive security strategies. Additionally, **dynamic role reassignment** enables swarm systems to recover from compromised agents by redistributing tasks and responsibilities across healthy robots.

AI agent systems, particularly those based on large-scale models like LLMs, require high computational resources and typically operate in highly sensitive, data-driven environments. Countermeasures for AI agents are designed around LLM security, **model integrity** and **data privacy**. Security strategies for these systems focus on model integrity, robustness against adversarial attacks, and data privacy. Techniques such as differential privacy, secure aggregation, and model-based verification are employed to mitigate risks of data leakage during training and inference. These countermeasures address inherent vulnerabilities in LLMs, including susceptibility to poisoning attacks, model inversion, and unauthorized data extraction.

4.4.2. Security design principles in swarm robots applicable to AI agents. The analysis of threats and countermeasures reveals that AI agents possess a broader attack surface, encompassing both physical and cyber domains. Nevertheless, the bio-inspired, resilient, and decentralized mechanisms developed in swarm robotics offer valuable insights that can strengthen the security and robustness of AI agent systems:

(a) **Redundancy and collective validation.** Swarm systems exhibit fault tolerance through functional redundancy, with multiple agents performing similar tasks independently. This concept can be adapted to AI systems using group verification or multi-agent consensus, enhancing robustness against hallucinations and adversarial attacks. For instance, instead of relying on a single LLM output, a swarm-inspired architecture could aggregate responses from multiple models and use majority voting or confidence-based filtering to ensure more reliable outcomes.

(b) **Decentralized trust and consensus.** In swarm robotics, no single agent holds complete control and decisions are emergent and consensus-driven. This principle could be translated into multi-agent AI systems for secure decision-making, particularly in federated or multi-agent RL environments.

(c) **Stigmergy-based security monitoring.** Swarm robots often exhibit stigmergic coordination, where actions are indirectly coordinated through environmental cues (e.g., pheromone trails). This mechanism can inspire implicit anomaly detection in AI agents by monitoring changes in shared environments or behavior patterns.

(d) **Self-repair and dynamic role redistribution.** Swarm systems can reassign roles dynamically when certain members fail or are compromised. This principle can guide the development of AI agents capable of adaptive role-switching, fallback strategies, or self-repair mechanisms in response to degraded performance or detected threats.

5. Challenges and future work

Based on the above analysis of security and privacy protection in the field of swarm robots and AI agents, this section mainly discusses the challenges and future work.

5.1. Challenges

5.1.1. Challenge 1: distributed and wireless communication security. Swarm robots and multi-agent systems rely on data exchange for collaborative decision-making. As a result, the data transmitted during communication is susceptible to communication attacks. This risk is particularly pronounced in open or untrusted network environments, where wireless communication systems are vulnerable to malicious interference, compromising data integrity, confidentiality, and authenticity.

While existing research has proposed various anomaly detection schemes and broadcast communication methods, designing efficient communication and encryption mechanisms remains a challenge. These mechanisms must ensure data confidentiality while accounting for low latency and the resource constraints of individual nodes in real-world applications.

Furthermore, DoS attacks and network congestion can prevent legitimate nodes within the system from communicating effectively. Addressing this issue requires traffic control and bandwidth allocation algorithms to mitigate DoS attacks, as well as rapid identification and filtering of malicious traffic under resource-constrained conditions. Additionally, dynamically adjusting communication paths and data transmission strategies to enhance the reliability and fault tolerance of communication links while ensuring efficient data transfer remains an open problem.

5.1.2. Challenge 2: the conflict between system collaboration and privacy protection. In both systems, there is an inherent conflict between data sharing and privacy protection. If the level of data sharing between system nodes/agents is too low, the system may fail to acquire sufficient global information, weakening collaborative decision-making and reducing overall performance. Conversely, excessive data sharing increases the risk of privacy leakage. Striking a balance between privacy protection and effective task collaboration is a core challenge in the privacy and security of multi-agent systems.

5.1.3. Challenge 3: tradeoff among security, privacy and efficiency. From the comparison of the countermeasures, we found that traditional data protection methods may have significant limitations. Inappropriate encryption mechanisms can introduce excessive computational overhead and storage costs, while differential privacy may negatively impact decision-making performance in real-time scenarios. Therefore, a major challenge lies in preserving data utility while minimizing the risk of privacy breaches, ensuring both secure and efficient collaboration within the system.

5.1.4. Challenge 4: risk brought from LLM. Integrating LLMs into multi-agent systems has become a popular trend due to their advanced capabilities in reasoning, decision-making, and communication. However, this also introduces several risks, including misinformation, privacy vulnerabilities, and security threats. LLMs can generate inaccurate or biased responses, sometimes producing misleading information that appears highly credible. Privacy concerns arise as they may memorize and inadvertently leak sensitive data, making them susceptible to attacks that extract confidential or proprietary information. Additionally, their black-box nature makes decision-making processes difficult to interpret, posing risks in critical applications like healthcare and finance. The substantial computational power required by LLMs also raises concerns about energy consumption, high operational costs, and the growing concentration of AI control among a few major entities. To ensure safer and more reliable deployment, ongoing research focuses on enhancing LLM robustness, privacy protection, explainability, and decentralized governance.

5.1.5. Challenge 5: human-machine collaboration and ethical issues. With the widespread application of robots, AI agent systems, their collaboration with human users has become increasingly common, especially in complex tasks where human-machine cooperation is essential for improving task execution efficiency. However, this growing interaction also raises ethical and societal concerns, particularly regarding decision-making transparency and the allocation of responsibility in cases of improper behavior. Addressing these issues not only requires advancements in related technologies but also necessitates compliance with ethical standards and legal regulations.

One major challenge is the difficulty in tracing the decision-making process, making it hard for humans to understand the reasoning behind their choices. Therefore, leveraging explainable AI techniques to enhance decision transparency, traceability, and intuitive interpretability is crucial. Additionally, when errors occur during task execution, it is often unclear whether the fault lies in technical failures, algorithmic design flaws, or human misuse. Ensuring that these systems adhere to controllability and ethical principles in critical decision-making, while using technological solutions to minimize the negative impact of unintended behaviors, remains a significant challenge.

5.2. Future work

Building on existing research and identified challenges, several promising directions for future work include, but are not limited to, multimodal fusion and intelligent collaboration, trustworthy and privacy-preserving solutions for multi-agent systems, and ethics and regulatory compliance. The following sections provide a detailed discussion of these potential research areas.

5.2.1. Multimodal fusion and intelligent collaboration. With the rapid advancement of multimodal sensing technologies, multi-agent systems are entering an era of highly integrated data fusion and intelligent task collaboration. This shift relies on advanced multimodal data processing and information fusion to enable comprehensive environmental perception and efficient decision-making. It is expected to drive wider and deeper applications across various domains, providing more

accurate, real-time, and intelligent solutions for complex scenarios. The key research directions include:

- **Deep integration and complementary use of multimodal data**: by overcoming the limitations of single-modal perception, data from multiple sensors (e.g., vision, speech, touch, temperature, and environmental monitoring) can be combined to enhance the system's overall sensing capabilities. While existing methods for multimodal data integration—such as feature concatenation and attention-based mechanisms—have shown promise, fully leveraging the richness of multimodal data remains a challenging task. Future research for multimodal fusion includes determining the confidence level of each modality, assessing inter-modal correlations, performing dimensionality reduction on high-dimensional multimodal features, and aligning data collected asynchronously across different modalities.
- **Cross-modal collaboration among system nodes**: system nodes can share and utilize different modalities of data in real time, improving coordination and collective intelligence through efficient information exchange. However, the heterogeneity of modalities across nodes poses significant challenges in constructing a unified feature space for effective collaboration. Additionally, it remains difficult to accurately weigh the contributions of different nodes during fusion and decision-making processes. A promising direction to address these issues is the integration of federated learning frameworks with game-theoretic approaches, which may help balance distributed workloads and optimize cross-modal collaboration in a decentralized manner.
- **Enhancing multimodal deep learning and intelligent inference capabilities**: the integration of multimodal data into meta-learning and reinforcement learning endows agents with enhanced perceptual and decision-making capabilities. However, policy optimization based on fused multimodal information presents significant challenges. Compared with single-modality inputs, multimodal inputs lead to higher-dimensional state spaces, more complex feature representations, and potential conflicts or redundancies between modalities. These factors make training stable and effective policies more difficult. A future direction lies in stage-wise learning to gradually integrate modalities, allowing the agent to incrementally learn meaningful features and dependencies across modalities. Moreover, enhancing the model's ability to generalize to out-of-distribution modalities or unseen modality combinations is crucial to avoid overfitting to specific training conditions and to ensure broader adaptability.

By integrating these three aspects, future multi-agent intelligent systems will play a more significant role in complex application scenarios, improving adaptability, robustness, and autonomy.

5.2.2. Trustworthy and privacy-preserving solutions for multi-agent systems. As data security and privacy protection become increasingly urgent concerns, future developments in trustworthy and privacy-preserving mechanisms for multi-agent systems will focus on three key areas:

- **Secure data sharing and privacy protection:** data is the foundation of multi-agent systems. Finding a balance between data availability and security to enable effective collaboration while minimizing privacy risks. However, achieving both privacy protection and high system performance presents a fundamental trade-off. A single approach is often insufficient to address all aspects of trust and privacy, necessitating a combination of multiple techniques such as federated learning, secure multi-party computation, differential privacy, trusted execution environments, and blockchain. This multi-layered and cross-domain security framework will help reconcile privacy protection with computation efficiency, secure data sharing, and system transparency, laying the foundation for a safer, more efficient, and trustworthy multi-agent ecosystem.
- **LLM security and reliability:** attacks such as prompt engineering and jailbreaking have become increasingly prevalent, posing significant challenges to the security and reliability of LLMs. These attacks exploit the inherent flexibility and openness of LLMs, allowing malicious actors to manipulate the models into generating harmful or unintended outputs. However, current countermeasures are often limited to specific models or attack types and are not universally applicable. To address this, there is a growing need for a more generalized framework for LLM security—one that can adapt to a wide range of potential vulnerabilities and attack vectors. Additionally, the integration of formal language verification methods is emerging as a promising trend.
- **Full-life cycle security:** with the rapid development of embodied intelligence, full-life cycle security has become essential to ensure safe and trustworthy interactions throughout the entire system process—from perception and decision-making to actuation and human-machine interaction. A comprehensive security framework should cover all critical layers, including the communication and infrastructure layer, the sensing layer, and the decision-making layer, addressing both cyber and physical threats across each stage of the system's operation.

5.2.3. Ethics and regulatory compliance. Human-machine interaction introduces higher requirements for ethic and regulation compliance, particularly as embodied intelligence enables machines to physically interact with humans. This includes the development of:

- Explainability **and transparency**: to enhance trust and controllability in human-machine collaboration, future systems will place greater emphasis on decision-making transparency and result explainability. Explainable AI techniques such as attention visualization, feature attribution (sharply value explanation, local interpretable model-agnostic explanations), and symbolic verification should be more deeply integrated into the inference process.

- **Fairness and non-discrimination**: ensuring fairness in AI systems is vital to prevent biased decision-making, especially in sensitive areas like healthcare and hiring. AI agent systems should implement **fairness-aware algorithms**, such as demographic parity and equal opportunity, to ensure equitable outcomes across different groups. **Data audits** and regular **bias monitoring** help detect and correct biases in training datasets, ensuring fairness over time. Additionally, fairness constraints should be embedded within the system to promote ethical decision-making, with metrics like group or individual fairness applied to evaluate and adjust the system's decisions.
- **Comprehensive countermeasures for ethical compliance:** a robust compliance strategy requires a multi-faceted approach to address ethical concerns throughout the AI lifecycle. This includes conducting **ethical reviews** at each stage of development, ensuring alignment with established ethical guidelines. Clear **responsibility attribution** should be defined for system designers, developers, and operators to ensure accountability. Ongoing **collaboration** between regulators, developers, and the public is essential to uphold ethical standards, ensuring that AI systems are transparent, fair, and ultimately serve the broader societal good.

6. Conclusion

In this paper, we presented a comprehensive survey of the security and reliability challenges in both swarm robotic systems and AI agent systems. By systematically reviewing the existing literature, we categorized security threats across the physical, communication, and application layers for both systems and examined the corresponding countermeasures. Our comparison and analysis highlight not only the strengths of existing solutions but also significant gaps, particularly in their adaptability and resilience to emerging threats.

Through our comparative analysis, we identified notable parallels in attack vectors and defense strategies between the two domains. These similarities suggest promising opportunities for cross-domain learning—where insights from the decentralized, resilient nature of swarm robotics can inform the development of more robust and secure AI agent architectures. Furthermore, we outlined the critical gaps in current research, particularly in terms of adaptive countermeasures, resilience against evolving attack vectors, and integration of the latest technological advancements in both fields.

By providing a unified analysis of the security challenges and solutions across these domains, we aim to contribute to the development of more secure, reliable, and robust intelligent systems. Our findings call for continued research focused on adaptive security solutions, cross-domain learning, and the refinement of system architectures to ensure the long-term stability and trustworthiness of both swarm robotic systems and AI agents.

Acknowledgment

This work is funded by an International Collaboration Fund for Creative Research of National Science Foundation of China (NSFC ICFCRT) under the Grant No. W2441019.

References

[1] Bonabeau E 1999 *Swarm Intelligence: From Natural to Artificial Systems* vol 2 (Oxford University Press) pp 25–34
[2] Brambilla M, Ferrante E, Birattari M and Dorigo M 2013 Swarm robotics: a review from the swarm engineering perspective *Swarm Intell.* **7** 1–41
[3] Creswell M, Shanahan M and Higgins I 2022 Selection-inference: exploiting large language models for interpretable logical reasoning (arXiv:2205.09712)
[4] Gong R *et al* 2023 Mindagent: emergent gaming interaction (arXiv:2309.09971)
[5] Julia W, Patrick M and Vladimir V, 2025 Agents *Google white paper*
[6] Howard D, Eiben A E, Kennedy D F, Mouret J B, Valencia P and Winkler D 2019 Evolving embodied intelligence from materials to machines *Nat. Mach. Intell.* **1** 12–19
[7] Wang Y, Pan Y, Su Z, Deng Y, Zhao Q, Du L, Luan T H, Kang J and Niyato D 2024 Large model agents: state-of-the-art, cooperation paradigms, security and privacy, and future trends (arXiv:2409.14457)
[8] The Robot Report Staff 2025 H2 clipper plans to deploy robotic swarms in aerospace manufacturing. *Expoloring the business and applications of robotics online*
[9] Winfield A F, Swana M, Ives J and Hauert S 2025 On the ethical governance of swarm robotic systems in the real world *Phil. Trans. A* **383** 20240142
[10] Vasu J 2025 Microsoft security microsoft unveils microsoft security copilot agents and new protections for AI *Microsoft Security Blog online*
[11] Plumb T 2025 Google cloud intros AI security agents, unified security platform to consolidate ops, triage, threat intel *VentureBeat online*
[12] Radanliev P, De Roure D, Maple C, Nurse J R, Nicolescu R and Ani U 2024 AI security and cyber risk in IoT systems *Front. Big Data* **7** 1402745
[13] Radanliev P, Santos O and Brandon-Jones A 2024 Capability hardware enhanced instructions and artificial intelligence bill of materials in trustworthy artificial intelligence systems: analyzing cybersecurity threats, exploits, and vulnerabilities in new software bills of materials with artificial intelligence *J. Defense Model. Simul.* **26**
[14] Hunt E R and Hauert S 2020 A checklist for safe robot swarms *Nat. Mach. Intell.* **2** 420–2
[15] Higgins F, Tomlinson A and Martin K M 2009 Survey on security challenges for swarm robotics *5th Int. Conf. on Autonomic and Autonomous Systems* (IEEE) pp 307–12
[16] Wilson J *et al* 2023 Trustworthy swarms *Proc. 1st Int. Symp. on Trustworthy Autonomous Systems* pp 1–11
[17] Andreoni M, Lunardi W T, Lawton G and Thakkar S 2024 Enhancing autonomous system security and resilience with generative AI: a comprehensive survey *IEEE Access* **12** 109470–93
[18] Xi Z *et al* 2025 The rise and potential of large language model based agents: a survey *Sci. China Inf. Sci.* **68** 121101

[19] Neupane S, Mitra S, Fernandez I and Saha S 2024 Security considerations in AI-robotics: a survey of current methods, challenges, and opportunities *IEEE Access* **12** 22072–97

[20] Lerman K, Martinoli A and Galstyan A 2005 A review of probabilistic macroscopic models for swarm robotic systems *Swarm Robotics: SAB 2004 Int. Workshop* pp 143–52

[21] Guo H, Meng Y and Jin Y 2010 Analysis of local communication load in shape formation of a distributed morphogenetic swarm robotic system *IEEE Congress on Evolutionary Computation* (IEEE) pp 1–8

[22] Dorigo M 1992 Optimization, learning and natural algorithms *PhD Thesis* Politecnico di Milano (https://doi.org/10.1177/112067219200200402)

[23] Kennedy J and Eberhart R 1995 Particle swarm optimization *Proc. ICNN'95th Int. Conf. on Neural Networks* vol 4 (IEEE) pp 1942–8

[24] Mirjalili S, Mirjalili S M and Lewis A 2014 Grey wolf optimizer *Adv. Eng. Softw.* **69** 46–61

[25] Auer L, Feichtner A, Steinhäusler F and Delleske R 2018 Swarm-technology for large-area photogrammetry survey and spatially complex 3D modelling *Int. J. Latest Res. Eng. Technol.* **4** 33–39

[26] Rubenstein M, Cornejo A and Nagpal R 2014 Programmable self-assembly in a thousand-robot swarm *Science* **345** 795–9

[27] Tosato P, Facinelli D, Prada M, Gemma L, Rossi M and Brunelli D 2019 An autonomous swarm of drones for industrial gas sensing applications *20th Int. Symp. on "A World of Wireless, Mobile and Multimedia Networks" (WoWMoM)* (IEEE) pp 1–6

[28] Hoar R, Penner J and Jacob C 2002 Evolutionary swarm traffic: if ant roads had traffic lights *Proc. 2002 Congress on Evolutionary Computation* (IEEE) vol 2 pp 1910–5

[29] Durante Z *et al* 2024 An interactive agent foundation model (arXiv:2402.05929)

[30] Figure AI 2024 (available at www.figure.ai/news/helix) Helix: a vision-language-action model for generalist humanoid control

[31] Zhu P, Zhou Z, Yao W, Dai W, Zeng Z and Lu H 2025 HI-GVF: shared control based on human-influenced guiding vector fields for human-multi-robot cooperation (arXiv:2502.11370)

[32] Zhang D, Feng G, Shi Y and Srinivasan D 2021 Physical safety and cyber security analysis of multi-agent systems: a survey of recent advances *IEEE/CAA J. Autom. Sin.* **8** 319–33

[33] Huang Q *et al* 2024 Position paper: agent AI towards a holistic intelligence (arXiv:2403.00833)

[34] Wu Q, Mei W and Zhang R 2019 Safeguarding wireless network with UAVs: a physical layer security perspective *IEEE Wirel. Commun.* **26** 12–18

[35] Yan C, Xu W and Liu J 2016 Can you trust autonomous vehicles: contactless attacks against sensors of self-driving vehicle *Def. Con.* **24** 109

[36] Fu Z, Zhi Y, Ji S and Sun X 2022 Remote attacks on drones vision sensors: an empirical study *IEEE Trans. Dependable Secure Comput.* **19** 3125–35

[37] Cozzolino D, Thies J, Rossler A, Niessner M and Verdoliva L 2021 SpoC: spoofing camera fingerprints *Proc. IEEE/CVF Conf. on Computer Vision and Pattern Recognition* pp 990–1000

[38] Yaacoub J-P A and Noura H N, Salman O and Chehab A 2022 Robotics cyber security: vulnerabilities, attacks, countermeasures, and recommendations *Int. J. Inf. Secur.* **21** 115–58

[39] Liu A *et al* 2024 Compromising embodied agents with contextual backdoor attacks (arXiv:2408.02882)

[40] Skorobogatov S 2009 Local heating attacks on flash memory devices *2009 IEEE Int. Workshop on Hardware-Oriented Security and Trust* pp 1–6

[41] Desnitsky V and Kotenko I 2021 Simulation and assessment of battery depletion attacks on unmanned aerial vehicles for crisis management infrastructures *Simul. Modelling Pract. Theory* **107** 102244

[42] Mekdad Y, Aris A, Babun L, Fergougui A E, Conti M, Lazzeretti R and Uluagac A S 2023 A survey on security and privacy issues of UAVs *Comput. Netw.* **224** 109626

[43] Vilches V M, Kirschgens L A, Calvo A B, Cordero A H, Vilches R I, Rosas D M, Mendia A M, Juan LUS G O, Ugarte I Z and Gil-Uriarte E 2021 Introducing the robot security framework (RSF), a standardized methodology to perform security assessments in robotics (arXiv:1806.04042)

[44] Pogliani M, Quarta D, Polino M, Vittone M, Maggi F and Zanero S 2019 Security of controlled manufacturing systems in the connected factory: the case of industrial robots *J. Comput. Virol. Hacking Tech.* **15** 161–75

[45] Zhang S, Pan Y, Liu Q, Yan Z, Choo K-K R and Wang G 2024 Backdoor attacks and defenses targeting multi-domain AI models: a comprehensive review *ACM Comput. Surv.* **57** 35

[46] Wang Y, Pan Y, Su Z, Deng Y, Zhao Q, Du L, Luan T H, Kang J and Niyato D 2025 Large model based agents: state-of-the-art, cooperation paradigms, security and privacy, and future trends (arXiv:2409.14457)

[47] Zhu Q, Rass S, Dieber B and Vilches V M 2021 Cybersecurity in robotics: challenges, quantitative modeling, and practice *ROB* **9** 1–129

[48] Higgins F, Tomlinson A and Martin K 2009 Threats to the swarm: security considerations for swarm robotics *Int. J. Inf. Secur.* **2** 288

[49] Shen X, Chen Z, Backes M, Shen Y and Zhang Y 2024 'Do anything now': characterizing and evaluating in-the-wild jailbreak prompts on large language models *Proc. on ACM SIGSAC Conf. on Computer and Communications Security CCS'24* pp 1671–85

[50] Khan R, Sarkar S and Mahata S K 2024 Security threats in agentic AI System (arXiv:2410.14728)

[51] Bouhata D, Moumen H, Mazari J A and Bounceur A 2022 Byzantine fault tolerance in distributed machine learning: a survey *J. Exp. Theor. Artif. Intell.* **37** 1–59

[52] Guerraoui R, Gupta N and Pinot R 2024 Byzantine machine learning: a primer *ACM Comput. Surv.* **56** 39

[53] Zhang G, Pan F, Mao Y, Tijanic S, Dang'ana M, Motepalli S, Zhang S and Jacobsen H-A 2024 Reaching consensus in the byzantine empire: a comprehensive review of BFT consensus algorithms *ACM Comput. Surv.* **56** 41

[54] Douceur J R 2002 *The Sybil Attack Peer-to-peer Systems* (*Lecture Notes in Computer Science* (Springer) vol 2429) pp 251–60

[55] Zhuo T Y, Li Z, Huang Y, Shiri F, Wang W, Haffari G and Li Y-F 2023 On robustness of prompt-based semantic parsing with large pre-trained language model: an empirical study on codex (arXiv:2301.12868)

[56] Zou W, Geng R, Wang B and Jia J 2024 PoisonedRAG: knowledge corruption attacks to retrieval-augmented generation of large language models (arXiv:2402.07867)

[57] Fang M, Cao X, Jia J and Gong N Z 2021 Local model poisoning attacks to byzantine-robust federated learning *29th USENIX security Symp.* pp 1605–22

[58] Liu Y *et al* 2024 Prompt injection attack against LLM-integrated applications (arXiv:2306.05499)

[59] Deng Z, Guo Y, Han C, Ma W, Xiong J, Wen S and Xiang Y 2025 AI agents under threat: a survey of key security challenges and future pathways *ACM Comput. Surv.* **57** 1–36

[60] Zhang S, Huang Y, Li W and Pan J 2025 Swarm robotic flocking with aggregation ability privacy *IEEE Trans. Autom. Sci. Eng.* **22** 10442–56
[61] Zheng H, Panerati J, Beltrame G and Prorok A 2020 An adversarial approach to private flocking in mobile robot teams *IEEE Robot. Autom. Lett.* **5** 1009–16
[62] Wang B and Gong N Z 2018 Stealing hyperparameters in machine learning *2018 IEEE Symp. on Security and Privacy (SP)* pp 36–52
[63] Morris J X, Kuleshov V, Shmatikov V and Rush A M 2023 Text embeddings reveal (almost) as much as text (arXiv:2310.06816)
[64] Petar R 2024 Digital security by design *Secur. J.* **37** 1640–79
[65] Petar R 2024 Integrated cybersecurity for metaverse systems operating with artificial intelligence, blockchains, and cloud computing *Front. Blockchain* **7** 1359130
[66] Wang Y and Ishii H 2020 Resilient consensus through event-based communication *IEEE Trans. Control Netw. Syst.* **7** 471–82
[67] Wen G, Lv Y, Zheng W X, Zhou J and Fu J 2023 Joint robustness of time-varying networks and its applications to resilient consensus *IEEE Trans. Autom. Control* **68** 6466–80
[68] Renganathan V and Summers T 2017 Spoof resilient coordination for distributed multi-robot systems *Int. Symp. on Multi-Robot and Multi-Agent Systems (MRS)* pp 135–41
[69] Wardega K, Tron R, von Hippel M, Nita-Rotaru C and Li W 2023 Byzantine resilience at swarm scale: a decentralized blocklist protocol from Inter-robot accusations *Proc. Int. Conf. on Autonomous Agents and Multiagent Systems* vol 9 pp 1430–8
[70] Yuan L and Ishii H 2021 Secure consensus with distributed detection via two-hop communication *Automatica* **131** 109775
[71] Gil S, Kumar S, Mazumder M, Katabi D and Rus D 2017 Guaranteeing spoof-resilient multi-robot networks *Auton. Robot.* **41** 1383–400
[72] Huang Y, Wang W, Jiang T and Zhang Q 2021 Detecting colluding sybil attackers in robotic networks using backscatters *IEEE/ACM Trans. Network.* **29** 793–804
[73] Shoukry Y, Mishra S, Luo Z and Diggavi S 2018 Sybil attack resilient traffic networks: a physics-based trust propagation approach *2018 ACM/IEEE 9th Int.Conf. on Cyber-Physical Systems (ICCPS)* pp 43–54
[74] Castelló Ferrer E 2019 The blockchain: a new framework for robotic swarm systems *Proc. Future Technologies Conf. (FTC)* vol 881 pp 1037–58
[75] Strobel V, Ferrer E C and Dorigo M 2018 Managing byzantine robots via blockchain technology in a swarm robotics collective decision making scenario *Proc. 17th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS'18)* pp 541–9
[76] Strobel V, Pacheco A and Dorigo M 2023 Robot swarms neutralize harmful byzantine robots using a blockchain-based token economy *Sci. Robot.* **8** 4636
[77] Strobel V, Castelló Ferrer E and Dorigo M 2020 Blockchain technology secures robot swarms: a comparison of consensus protocols and their resilience to byzantine robots *Front. Robot. AI* **7** 54
[78] Pacheco A, Strobel V and Dorigo M 2020 A blockchain-controlled physical robot swarm communicating via an ad-hoc network *Int. Conf. on Swarm Intelligence* pp 3–15
[79] Jones A and Straub J 2017 Using deep learning to detect network intrusions and malware in autonomous robots *Cyber Sens.* **10185** 45–50
[80] Abouelyazid M 2023 Adversarial deep reinforcement learning to mitigate sensor and communication attacks for secure swarm robotics *Intell. Connect. Emerg. Technol.* **8** 94–112
[81] Masadeh A, Alhafnawi M, Salameh H A B, Musa A and Jararweh Y 2024 Reinforcement learning-based security/safety UAV system for intrusion detection under dynamic and uncertain target movement *IEEE Trans. Eng. Manage.* **71** 12498–508
[82] Shafarenko A 2024 A zero-trust swarm security architecture and protocols *Cryptology ePrint Archive*
[83] Cogliani S, Feng B, Ferradi H, Géraud R, Maimuţ D, Naccache D, Portella do Canto R and Wang G 2018 Public key-based lightweight swarm authentication *Cyber-Physical Systems Security* (Springer) pp 255–67
[84] Maimuţ D and Teşeleanu G 2020 *A Generic View on the Unified Zero-knowledge Protocol and Its Applications* (*Lecture Notes in Computer Science* (Cryptology ePrint Archive)) pp 32–46
[85] George T 2021 Lightweight swarm authentication *Innovative Security Solutions for Information Technology and Communications: 14th Int. Conf., SecITC 2021* **13195** 248–59
[86] Ruan M, Gao H and Wang Y 2019 Secure and privacy-preserving consensus *IEEE Trans. Autom. Control* **64** 4035–49
[87] Kishida M 2018 Encrypted average consensus with quantized control law *IEEE Conf. on Decision and Control* pp 5850–6
[88] Gojković M and Schranz M 2024 Preserving privacy in logistics by using swarm intelligence from the bottom-up *2024 IEEE 12th Int. Conf. on Intelligent Systems (IS)* pp 1–7
[89] Wang X, He J, Cheng P and Chen J 2018 Privacy preserving average consensus with different privacy guarantee *2018 Annual American Control Conf. (ACC)* pp 5189–94
[90] Chen W, Wang Z, Hu J and Liu G-P 2023 Differentially private average consensus with logarithmic dynamic encoding–decoding scheme *IEEE Trans. Cyber.* **53** 6725–36
[91] Zhang Z, Zhu H and Xie M 2024 Differential privacy may have a potential optimization effect on some swarm intelligence algorithms besides privacy-preserving *Inf. Sci.* **654** 119870
[92] Ramos G and Pequito S 2023 Designing communication networks for discrete-time consensus for performance and privacy guarantees *Syst. Control Lett.* **180** 105608
[93] Zhang J, Lu J and Hadjicostis C N 2024 Average consensus for expressed and private opinions *IEEE Trans. Autom. Control* **69** 5627–34
[94] Torra V, Galván E and Navarro-Arribas G 2022 PSO + FL = PAASO: particle swarm optimization + federated learning = privacy-aware agent swarm optimization *Int. J. Inf. Secur.* **21** 1349–59
[95] Zhou X, Liang W, Wang K I-K, Yan Z, Yang L T, Wei W, Ma J and Jin Q 2023 Decentralized P2P federated learning for privacy-preserving and resilient mobile robotic systems *IEEE Wirel. Commun.* **30** 82–89
[96] Wang J, Wang J, Zhang F, Li J, Li Z and Chen T 2025 Feature map poisoning attack and dual defense mechanism for federated prototype learning *J. Softw.* **36** 1355–74
[97] Domingo-Ferrer J, Blanco-Justicia A, Manjon J and Sanchez D 2022 Secure and privacy-preserving federated learning via co-utility *IEEE Int. Things J.* **9** 3988–4000
[98] Narajala V S and Habler I 2025 Enterprise-grade security for the model context protocol (mcp): frameworks and mitigation strategies (arXiv:2504.08623)
[99] Zeng Y, Wu Y, Zhang X, Wang H and Wu Q 2024 AutoDefense: multi-agent LLM defense against jailbreak attacks *Neurips Safe Generative AI Workshop*
[100] Wang B, Yao Y, Shan S, Li H, Viswanath B and Zheng H 2019 Neural cleanse: identifying and mitigating backdoor

attacks in neural networks *2019 IEEE Symp. on Security and Privacy (SP)* pp 707–23
[101] Manakul P, Liusie A and Gales M 2023 SelfCheckGPT: zero-resource black-box hallucination detection for generative large language models *Proc. Conf. on Empirical Methods in Natural Language Processing* pp 9004–17
[102] Xu C, Wang J, Guzmán F, Rubinstein B and Cohn T 2021 Mitigating data poisoning in text classification with differential privacy *Findings of the Association for Computational Linguistics: EMNLP 2021* pp 4348–56
[103] Dai J, Pan X, Sun R, Ji J, Xu X, Liu M, Wang Y and Yang Y 2024 Safe RLHF: safe reinforcement learning from human feedback *12th Int. Conf. on Learning Representations (ICLR)*
[104] Islam N T, Karkevandi M B and Najafirad P 2024 Code security vulnerability repair using reinforcement learning with large language models (arXiv:2401.07031)
[105] Hu J, Tao L, Yang J and Zhou C 2023 Aligning language models with offline learning from human feedback (arXiv:2308.12050)
[106] Liu F, Lin K, Li L, Wang J, Yacoob Y and Wang L 2024 Mitigating hallucination in large multi-modal models via robust instruction tuning *12th Int. Conf. on Learning Representations (ICLR)*
[107] He Z, Li Z, Yang S, Qiao A, Zhang X, Luo X and Chen T 2024 Large language models for blockchain security: a systematic literature review (arXiv:2403.14280)
[108] Jones E, Dragan A, Raghunathan A and Steinhardt J 2023 Automatically auditing large language models via discrete optimization *Proc. 40th Int. Conf. on Machine Learning (ICML)* pp 15307–29
[109] Balunovic M, Beurer-Kellner L, Fischer M and Vechev M 2024 AI agents with formal security guarantees *ICML 2024 Next Generation of AI Safety Workshop*
[110] Chai Y, Sun H, Fang H, Wang S, Sun Y and Wu H 2025 MA-RLHF: reinforcement learning from human feedback with macro actions *13th Int. Conf. on Learning Representations (ICLR)*
[111] Gorbatovski A, Shaposhnikov B, Malakhov A, Surnachev N, Aksenov Y, Maksimov I, Balagansky N and Gavrilov D 2024 Learn your reference model for real good alignment (arXiv:2404.09656)
[112] Zuo X, Wang M, Zhu T, Zhang L, Ye D, Yu S and Zhou W 2024 Federated trustchain: blockchain-enhanced LLM training and unlearning (arXiv:2406.04076)
[113] Behnia R, Ebrahimi M R, Pacheco J and Padmanabhan B 2022 EW-tune: a framework for privately fine-tuning large language models with differential privacy *2022 IEEE Int. Conf. on Data Mining Workshops (ICDMW)* pp 560–6
[114] Mai P, Yan R, Huang Z, Yang Y and Pang Y 2024 Split-and-denoise: protect large language model inference with local differential privacy *Proc. 41st Int. Conf. on Machine Learning (ICML)* pp 34281–302
[115] Chua L, Ghazi B, Huang Y, Kamath P, Kumar R, Liu D, Manurangsi P, Sinha A and Zhang C 2024 Mind the privacy unit! user-level differential privacy for language model fine-tuning *1st Conf. on Language Modeling*
[116] Chen Y, Li T, Liu H and Yu Y 2023 Hide and seek (has): a lightweight framework for prompt privacy protection (arXiv:2309.03057)
[117] Demir M M, Otal H T and Canbaz M A 2025 LegalGuardian: a privacy-preserving framework for secure integration of large language models in legal practice (arXiv:2501.10915)
[118] Frikha A, Walha N, Nakka K K, Mendes R, Jiang X and Zhou X 2024 IncogniText: privacy-enhancing conditional text anonymization via LLM-based private attribute randomization *Neurips Safe Generative AI Workshop 2024*
[119] Zhang S, Yi X, Xing H, Ye L, Hu Y and Li H 2024 Adanonymizer: interactively navigating and balancing the duality of privacy and output performance in human-LLM interaction (arXiv:2410.15044)
[120] Chen C, Feng X, Zhou J, Yin J and Zheng X 2023 Federated large language model: a position paper (arXiv:2307.08925)
[121] Sani L *et al* 2024 The future of large language model pre-training is federated *Int. Workshop on Federated Foundation Models in Conjunction with NeurIPS 2024*
[122] Ye R, Wang W, Chai J, Li D, Li Z, Xu Y, Du Y, Wang Y and Chen S 2024 OpenFedLLM: training large language models on decentralized private data via federated learning *Proc. 30th ACM SIGKDD Conf. on Knowledge Discovery and Data Mining* vol 33 pp 6137–47
[123] Zheng J, Zhang H, Wang L, Qiu W, Zheng H and Zheng Z 2024 Safely learning with private data: a federated learning framework for large language model (arXiv:2406.14898)
[124] Ezzeldin Y H, Yan S, He C, Ferrara E and Avestimehr S 2023 Fairfed: enabling group fairness in federated learning *In Proc. AAAI Conf. on Artificial Intelligence* 37 pp 7494–502
[125] Yang Y, Payani A and Naghizadeh P 2024 Enhancing group fairness in federated learning through personalization (arXiv:2407.19331)
[126] Rathee D, Li D, Stoica I, Zhang H and Popa R 2024 Mpc-minimized secure LLM inference (arXiv:2408.03561)
[127] Li S, Yao L, Gao J, Zhang L and Li Y 2024 Double-I watermark: protecting model copyright for LLM fine-tuning (arXiv:2402.14883)
[128] Habler I, Huang K, Narajala V S and Kulkarni P 2025 Building a secure agentic AI application leveraging A2A protocol (arXiv:2504.16902)
[129] Wang X, Jiang H, Yu Y, Yu J, Lin Y, Yi P, Wang Y, Qiao Y, Li L and Wang F-Y 2024 Building intelligence identification system via large language model watermarking: a survey and beyond (arXiv:2407.11100)

Yuping Yan received her PhD in the Department of Informatics from Eötvös Loránd University. She is currently a postdoctoral researcher at the Trustworthy and General AI Laboratory, School of Engineering, Westlake University, Hangzhou, China. Her research interests include privacy-preserving techniques, LLM-safety, and federated learning.

Yuhan Xie is currently an undergraduate student at the School of Cyber Engineering, Xidian University and a visiting student at the Trustworthy and General AI Laboratory, Westlake University. Her research interests include embodied intelligent security and LLM safety.

Junfeng Tang received BSc and MSc degrees in Intelligent Science and Technology and Computer Science and Technology from Xidian University, Xi'an, China in 2021 and 2024, respectively. Currently, he is a PhD degree candidate student at the School of Engineering, Westlake University, Hangzhou, China. His current research interests include optimization, robotic manipulation, and machine learning.

Yuanshuai Li is currently an undergraduate student at Nantong University, pursuing a degree in computer science. His research interests include federated learning, trustworthy optimization, and privacy-preserving techniques.

Yaochu Jin (Fellow, IEEE) received BSc, MSc, and PhD degrees in automatic control from Zhejiang University, Hangzhou, China, in 1988, 1991, and 1996, respectively, and the Dr-Ing degree in neuroinformatics from Ruhr-University Bochum, Bochum, Germany, in 2001.

He is presently Chair Professor for AI with the School of Engineering, Westlake University, Hangzhou, China, leading the Trustworthy and General AI Laboratory. He is the recipient of the 2025 IEEE Frank Rosenblatt Award for contributions to evolutionary optimization of complex systems. He was an Alexander von Humboldt Professor for AI with the Faculty of Technology, Bielefeld University, Bielefeld, Germany, and Surrey Distinguished Chair of Computational Intelligence with the Department of Computer Science, University of Surrey, Guildford, UK His main research interests include data-driven evolutionary optimization, trustworthy machine learning, multi-objective evolutionary learning, and evolutionary developmental systems.