

A Concern-Oriented Governance Approach for Securing Agentic AI

Sandeep Gupta

s.gupta@qub.ac.uk

Queen's University Belfast

Marco Bombieri

University of Trento

Bruno Crispo

University of Trento

Paolo Giorgini

University of Trento

Research Article

Keywords: Agentic AI, Cybersecurity, Large Language Models, Trustworthy AI, Governance

Posted Date: June 11th, 2026

DOI: <https://doi.org/10.21203/rs.3.rs-9632595/v1>

License: © ⓘ This work is licensed under a Creative Commons Attribution 4.0 International License.

[Read Full License](#)

Additional Declarations: No competing interests reported.

A Concern-Oriented Governance Approach for Securing Agentic AI

Sandeep Gupta^{1*}, Marco Bombieri², Bruno Crispo²,
Paolo Giorgini²

^{1*}Centre for Secure Information Technologies (CSIT), Queen's
University Belfast, UK.

²Department of Information Engineering and Computer Science,
University of Trento, Italy.

*Corresponding author(s). E-mail(s): s.gupta@qub.ac.uk;
Contributing authors: marco.bombieri@unitn.it; bruno.crispo@unitn.it;
paolo.giorgini@unitn.it;

Abstract

Agentic Artificial Intelligence (AgAI) embodies a substantial progression beyond predictive and generative AI, facilitating autonomous decision-making, planning, tool use, and the execution of complex, multi-step tasks with minimal human intervention across diverse domains. This increased autonomy and capability, however, also expand the potential impact and risk surface of these systems. Consequently, attacks on AgAI could lead to severe consequences, jeopardizing human lives, financial stability, and overall well-being. Thus, adherence to fundamental security properties, namely, *Confidentiality*, *Integrity*, and *Availability* (CIA), is critical to ensuring the safe, reliable, and trustworthy deployment of AgAI. In this paper, we first identify vulnerabilities and attack surfaces in AgAI by formalizing a four-layer architecture (orchestration, perception, reasoning, and action) together with the explicit states and transitions of task execution. We subsequently map eleven primary attack classes to the three core surfaces, i.e., *data*, *models*, and *infrastructure*, and to the specific AgAI layer where each attack is most likely to occur. Building on this mapping, we provide a detailed examination of concrete strategies and techniques for each surface. Finally, we introduce a Concern-Approach Matrix that links AgAI-specific risks to targeted countermeasures across security, AI/ML risk management, privacy/compliance, and people/organization dimensions, complemented by a lifecycle-to-layer operational checklist that translates the matrix into actionable defense mechanisms. The

resulting framework offers designers, operators, and policymakers a structured, architecture-driven approach to AgAI security and governance.

Keywords: Agentic AI, Cybersecurity, Large Language Models, Trustworthy AI, Governance

1 Introduction

Agentic Artificial Intelligence (AgAI) represents a significant advancement over traditional AI paradigms, including predictive and generative AI [1]. Predictive AI focuses on analyzing data using machine learning to forecast outcomes, behaviors, or trends based on historical patterns [2]. Generative AI, while also fundamentally predictive in nature (such as by predicting the next token, pixel, or sample given a context), extends this capability to create new and original content, including text, images, video, and audio [3]. As autonomous software entities, AgAI systems transcend these paradigms by facilitating autonomous decision-making, planning, tool use, and the execution of complex, multi-step tasks across dynamic environments with minimal human intervention. This evolution has unlocked new avenues for innovation across virtually every domain, contributing to improvements in overall quality of life. Emerging agentic AI platforms include Google Agent Development Kit (ADK) [4] for building multi-agent systems, Anthropic Claude Agent SDK [5] for constructing autonomous tool-using agents, and enterprise-grade environments such as Microsoft Copilot Studio [6], OpenAI agent frameworks [7], and IBM AI agent platforms [8].

Leading organizations are already deploying AgAI for autonomous process and delivery management (e.g., dynamic supply chain optimization and automated reordering), customer service, finance reconciliation, logistics routing, and medical diagnostics. Retailers such as Amazon and Walmart are integrating robotics agentic AI-enabled automation throughout their supply chains [9, 10], while Mercedes-Benz has embedded Google Cloud’s Automotive AI Agent in its MBUX Virtual Assistant for real-time personalized navigation and information [11]. Bayer leverages AgAI for forecasting cold and flu outbreaks [12]. The utilization of AgAI is rapidly progressing from routine tasks to high-risk, safety-critical systems such as autonomous driving, banking, and healthcare. However, this transformative potential also introduces substantial risks. Gartner [13] predicts that by 2028, 25% of enterprise security breaches will originate from the malicious use of AI agents, perpetrated by both insider threats and external attackers. When compromised, AgAI systems entrusted with critical infrastructure or high-stakes operations can exacerbate existential threats and create new, unforeseen vulnerabilities potentially, endangering lives, finances, and public safety.

This paper presents a thorough investigation of the security requirements for an AgAI system, focusing on confidentiality, integrity, and availability (CIA), which are essential for successful deployment, reliable operation, and widespread user acceptance. A typical AgAI system can comprise of perception (data acquisition), reasoning (cognitive processing and decision-making), action (task execution), and orchestration (workflow management and coordination) layers. We categorize and analyze potential

attack vectors, strategies, and techniques targeting data, models, and other critical resources within each layer of AgAI systems. For instance, attacks such as evasion [14] and instruction-tuning [15] largely affect the perception layer. At the reasoning layer, models are susceptible to data poisoning [16, 17]. Significant threats to model security also include model extraction [18], logic corruption [19], and membership inference [20]. Data exfiltration [21, 22] poses a pervasive threat, with the potential to compromise data across the reasoning, action, and orchestration layers. Moreover, resource manipulation [23], denial-of-service (DoS) [17], man-in-the-middle (MitM) [24], and side-channel attacks [25] represent critical risks that can adversely impact all four layers of AgAI systems.

1.1 Contributions

The main contributions of the paper are outlined as follows:

- Present an attack mapping that establishes a rigorous foundation for identifying and analyzing vulnerabilities, threats, and attack surfaces across the entire AgAI workflow. This mapping is derived using a four-layer architecture for AgAI systems, comprising the orchestration, perception, reasoning, and action layers, together with task execution states and transitions.
- Delineate the security requirements of AgAI systems by mapping the eleven primary attack classes to the three core attack surfaces (data, models, and infrastructure) and to the specific AgAI layer with the highest likelihood of occurrence, supported by a comprehensive examination of concrete strategies and techniques for each surface.
- Propose a Concern-Approach Matrix that systematically maps AgAI-specific risks and concerns to tailored countermeasures across four dimensions (security, AI/ML risk management, privacy/compliance, and people/organization), complemented by a lifecycle-to-layer operational checklist that translates the matrix into practical, implementable defense mechanisms.

1.2 Paper structure

The remaining sections are organized as follows: Section 2 provides a comparison with similar initiatives in the AgAI security literature, highlighting the gaps addressed by the present work. Section 3 introduces the conceptual foundations of AgAI systems, presents the proposed four-layer reference architecture, the states and transitions of task execution, and the layer-wise mapping of primary attacks. Section 4 provides a detailed examination of strategies and techniques for attacking AgAI systems across the three primary attack surfaces (data, models, and infrastructure). Section 5 presents the proposed security and governance framework, including the Concern-Approach Matrix and the corresponding operational checklist derived from lifecycle-to-layer mappings. Finally, Section 6 presents the conclusions of the paper and outlines directions for future work. A provides an overview of foundational security frameworks (Microsoft and Google).

2 A comparison with similar initiatives

Since 2023, the literature around AgAI has expanded rapidly, driven by the emergence of LLM-based agents capable of maintaining long-lived state through memory stores, files, vector databases, invoking external tools (APIs, plugins, code execution), and autonomously planning and acting across multiple steps. Compared to standalone chat assistants, these architectural additions create new attack surfaces and compound risks: untrusted inputs can be transformed into tool calls, tool outputs can be re-ingested as trusted context, and retrieved memories can steer both reasoning and actions. Finally the reasoning component can be jailbreak. To order the landscape around AgAI and security, literature offer several reviews that can be grouped into: (1) **surveys/taxonomies** that organize threats and defenses across agent pipelines, and (2) **systematization and benchmarks** that formalize and empirically evaluate concrete attacks, e.g., indirect prompt injection, and memory or retrieval poisoning.

- *Surveys and taxonomies for AgAI security*: The authors of [26] provide a survey of security and privacy issues in AgAI, introducing background on agent construction and organizing the threat landscape, impacts, and defense strategies, supported by case studies. In particular, they recognize that some vulnerabilities are inherited from the underlying LLM model (e.g., hallucinations, catastrophic forgetting, language misunderstanding, and inference attacks), while others are specific threats to the agentic architecture (e.g., knowledge poisoning, output, and functional manipulation). In [27], the authors offer a comprehensive survey emphasizing a modular view of AgAI and propose a component-centric classification of attacks and defenses and evaluation criteria (e.g., effectiveness, cost, and generality), while also discussing how they can be used in security contexts. They recognize that some attacks are data-targeting, others are model-targeting, and others target the multi-agent communication mechanism. In [28], they broaden the scope beyond prompt- and model-level threats by explicitly addressing workflow and protocol vulnerabilities in LLM-agent ecosystems. Their survey frames an end-to-end threat model spanning host-to-tool and agent-to-agent communication, and relates the taxonomy to real-world vulnerability examples, such as prompt-to-SQL injections and the toxic agent flow.
- *Formalizations and benchmarks for prompt injection and tool surfaces*: Prompt injection remains the most widely studied class of attacks for LLM-integrated applications. In [29], the authors propose a formal framework for prompt injection and benchmark multiple attacks and defenses across tasks and model backbones, enabling quantitative comparisons. In [30], they measure the social and ethical effects that such attacks can bring. [31] extends the analysis by introducing a benchmark for indirect prompt injection attacks specific to AgAI systems that consume untrusted artifacts. Their results demonstrate that popular agent designs remain vulnerable to instruction smuggling and data exfiltration behaviors, highlighting the need for component-aware mitigations at the perception and tool boundaries.
- *Retrieval and memory attack surfaces*: Many agents rely on retrieval-augmented generation (RAG) or long-term memory to ground reasoning, but this external knowledge introduces a distinct and practical attack surface. In [32], they demonstrate knowledge corruption attacks against RAG by inserting a small number of

malicious texts into a knowledge base, steering the model toward attacker-chosen targets and showing that common defenses can be insufficient. In AgAI settings, this risk is amplified because retrieved content can influence not only the final response but also planning and subsequent tool actions.

- *Agent-level security evaluation beyond a single vector*: While prompt injection is central, agentic systems expose additional stages (memory retrieval, planning traces, tool invocation, and multi-step execution). [33] extends the previously mentioned papers by introducing an evaluation framework that spans multiple agent scenarios, tool ecosystems, and LLM backbones. Importantly, it reports vulnerabilities across distinct operational stages (e.g., system prompt, user prompt handling, tool usage, and memory retrieval) and evaluates attacks and defenses beyond classic prompt injection, such as memory poisoning and backdoor-style behaviors.
- *Gap analysis and positioning*: Existing surveys provide broad catalogs and useful taxonomies, but this paper extends them by (i) explicitly grounding threats in a reference agent architecture, (ii) mapping attacks to concrete AgAI components (prompting, memory, tool interfaces, protocol/connector layers), and (iii) providing actionable guidance for designers and operators to mitigate such effects.

While, benchmarks and systematization deliver empirical evidence, they often focus on a subset of the pipeline (e.g., prompt-level or tool-level vulnerabilities) and typically stop short of linking technical controls to higher-level engineering concerns and operational ownership. In contrast, this paper complements prior works by adopting an explicitly architecture-driven perspective: given a baseline AgAI architecture, we systematically map common attacks to the specific components they target (model/prompting, perception inputs, memory/RAG, orchestration and tool interfaces, and protocol/connectors). Building on this mapping, we consolidate mitigation strategies and connect them to broader design and governance concerns via a concern–approach matrix, aiming to support both design-time reasoning and operational decision making.

3 Agentic AI systems

Multiple definitions of AgAI currently exist in the literature. It is typically conceptualized as a coordinated ecosystem of agents, in which specialized components share memory, distribute responsibilities, and collaborate to solve complex problems by making decisions and taking actions over time, rather than merely responding to isolated prompts or commands [34]. As illustrated in Table 1, AgAI is broadly not limited to LLM-based agents, but instead refers to any AI system exhibiting goal-directed, autonomous, and adaptive behavior. Whereas, an AI agent is typically defined as an autonomous software entity designed to *perceive* its environment, *reason* over information, make decisions, and *act* to achieve specific goals with limited human intervention.

Prior studies [1, 35] identify the compromise of AgAI systems as a significant security risk, particularly when such systems manage critical infrastructure or perform context-sensitive operations, where breaches could result in widespread harm. However, given AgAI’s transformative potential across diverse domains and the lack of

Table 1 Gen AI vs. AI Agent vs. Agentic AI

Terminology	Capability Scope	and	Tool Usage	Autonomy Behavior	and	Human-in-the-Loop
Generative AI (LLM / VLM / MLLM)	Generates text, images, audio, and video depending on modality. Responds based on pretrained knowledge and in-context input.		None	Reactive only - produces output in response to a prompt; does not initiate or execute actions		Continuous - every output is directly prompted and reviewed by a human
AI Agent	Perceives inputs, reasons over them, and executes discrete tasks through tool calls		Access to tools (APIs, databases, code execution, etc.)	Single-goal directed; selects and executes actions to complete a specified task		Partial - human defines the goal and reviews outcomes; agent acts autonomously within that scope
Agentic AI	Pursues complex, multi-step goals through planning, sub-task decomposition, and coordination of multiple agents or tools		Access to tools and other agents	Goal-directed, persistent, and adaptive; operates with minimal human intervention across extended task horizons		Minimal to none - human sets high-level objectives; system plans, decides, and acts with little or no per-step oversight

standardized security and governance frameworks, a rigorous examination of requirements is necessary to ensure safe deployment, reliable operation, and broad societal acceptance.

3.1 Attacks targeted AgAI

Table 2 presents instances of cyberattacks against AgAI systems observed in real-world incidents, red-teaming assessments, and proof-of-concept (PoC) demonstrations. Collectively, these instances emphasize the importance of maintaining adherence to the CIA triad, as violations can precipitate undesirable consequences including data breaches, operational disruptions, and safety hazards. To enable a thorough analysis of these attack vectors and their points of manifestation, we trace the attacks back to a baseline architectural framework of AgAI systems presented in Section 3.2¹.

Table 2: Attack incidents on agentic AI systems.

Case	Domain	Layer(s) Affected	Attack Type	Surface/ Impact on CIA	Evidence/Source
Adversa Report: Agentic failures in crypto/supply chain [36]	AI Financial/ Logistics	Reasoning, Action	Data (Poisoning 03), Models (Goal Manipulation 08)	Integrity (Theft/Fraud), Availability (Service Outages)	70% of 2025 incidents involved GenAI but agentic systems caused crypto thefts (2.3M USD loss) and API abuses 46.6% jailbreak success rate.

¹The colored markers in Table 2 correspond to the attack categories defined in Figure 2 and can be interpreted as 01-Evasion, 02-Instruction tuning, 03-Data poisoning, 04-Data exfiltration, 05-Model extraction, 06-Logic corruption, 07-Membership inference, 08-Resource manipulation, 09-DoS, 10-MitM, and 11-Side-channel attacks.

Anthropic Claude Hacking Operation [37]	Vibe	Cybersecurity (multi-sector: government, healthcare, finance)	Reasoning, Action	Models/Agents Misuse (automated reconnaissance, code execution, extortion 08)	Confidentiality (personal / financial data theft from 17 organization), Integrity (targeted extortion up to 500K USD), Availability (network penetration/disruption)	Anthropic Threat Report, Aug 2025
Claude Vulnerabilities (CVE-2025-54794/54795) [38]	Code	Software Development	Action	Infrastructure/Command Injection 01, Path Traversal	Confidentiality (unauthorized file access), Integrity (bypassed restrictions via AI-assisted reverse-engineering)	NIST database, Apr 2025
Driverless vulnerabilities [39]	car	Autonomous Driving	Perception, Action	Data (Adversarial Examples 01), Infrastructure (Side-Channel 01)	Integrity (Behavioral Manipulation), Availability (Collision Risk)	PoC demos: Malicious inputs caused system failures in 25% of tests, 3979 AV incidents (2021-2024), 15% AI-related.
EchoLeak (CVE-2025-32711) in Microsoft Copilot [40]		Enterprise Productivity in 365	Perception, Reasoning	Data (Prompt Injection 02), Infrastructure (Zero-Click)	Confidentiality (Data Exfiltration), Integrity (Unauthorized Access)	Zero-click vulnerability allowing hackers to access user data without interaction; patched in June 2025.
Model namespace reuse in Vertex AI [41]	AI	Model Deployment/Cloud AI	Reasoning, Orchestration	Models (supply chain poisoning 03), Infrastructure (remote code execution (RCE) via abandoned namespaces 08)	Confidentiality (Infrastructure access and data exposure), Integrity (Model substitution), Availability (Automated deployment disruptions)	PoC: reverse shell models executed in GCP containers post-namespaces hijacking, 100% success in controlled tests disclosed Sep 2025.
OpenAI Internal Design Theft [42]	AI Research		Reasoning	Internal 05	Confidentiality (Stolen AI technology designs/discussions)	Lawsuit, Aug 2024.
Palo Alto 42 scenarios [43]	Unit agentic	Multi-Domain (e.g., Logistics APIs)	All Layers	Infrastructure (DoS 06/MitM 01), Models (Extraction 05)	Availability (Resource Exhaustion), Confidentiality (Exfiltration)	Nine PoC attacks, e.g., tool misuse in logistics APIs led to simulated 500K USD supply chain disruption.
RCE AI Framework [44]	in AI Agent Development		Action	Models/Infrastructure, RCE Vulnerability 09	Integrity (potential code execution leading to model tampering), Availability (disrupted agent operations)	IBM X-Force Threat Intelligence Index, 2025.

Salesforce Agentforce ForcedLeak [45]	Finance/CRM (Sales and Customer Data Management)	Perception, Reasoning	Data (Indirect Prompt Injection Models Manipulation (Agent 06)), (Unauthorized Access/Exfiltration Infrastructure (Unsecured database mis-configuration 09))	Confidentiality (PII and Sales Data Leakage), Integrity (CRM Data Alteration/Deletion)	PoC via XSS in Web-to-Lead forms, tricked agents into exfiltrating customer leads/emails to attacker domains obtaining 100% success in tests, disclosed Sep 2025.
Serviceaide Agentic AI Breach [46]	Medical/Healthcare Data (Patient Data Management and Diagnostics)	Perception, Reasoning	Data (Unauthorized Access/Exfiltration Infrastructure (Unsecured database mis-configuration 09))	Confidentiality (PHI Exposure of 483,126 Patients), Integrity (Potential Data Tampering)	Autonomous AI database left exposed without access controls, enabling breach of protected health information confirmed incident via regulatory filings, May 2025.
SharePoint exploitation via Copilot agents [47]	Enterprise Collaboration	Action, Orchestration	Models (Tool Misuse Infrastructure (Privilege Escalation 09))	Integrity (Malicious Actions), Availability (Workflow Disruption)	Red-team PoC: Attackers exploited agents to exfiltrate files, success rate 40% in simulated engagements.

3.2 Baseline architecture

As illustrated in Figure 1, we derive a four-layer baseline architecture for AgAI comprising *orchestration*, *perception*, *reasoning*, and *action*. The orchestration layer manages workflow execution by assigning tasks, optimizing resource allocation, and facilitating inter-agent communication. It serves as a strategic control and coordination hub that orchestrates collaboration among agents to achieve complex goals, ensuring reliable, adaptive, and well-governed execution across dynamic workflows. The perception layer initiates task processing once tasks are scheduled by the orchestration layer. It acquires external and contextual data from sensors, databases, and digital interfaces, as applicable [48].

The reasoning layer, the core intelligence layer responsible for cognitive processing and decision-making, comprises a Task Planning Agent (TPA) and a Task Setup Agent (TSA). The TPA integrates large vision-language and multimodal foundation models and incorporates reflective processes for meta-reasoning and self-critique during cognition, alongside long-term memory (semantic/episodic) and feedback-loop mechanisms for task planning. Models such as GPT-4, Gemini, LLaVA, and Llama demonstrate broad understanding and are capable of performing complex reasoning tasks [49]. Reflection refers to an agent’s ability to analyze its own decision-making processes and correct errors through meta-reasoning, effectively engAgAIing in self-inspection that enhances autonomy and enables higher-level problem solving [50]. Both internal sources (e.g., self-reflection) and external sources (e.g., task-execution outcomes from the action layer, environmental observations, and inter-agent signals) can be leveraged during reasoning.

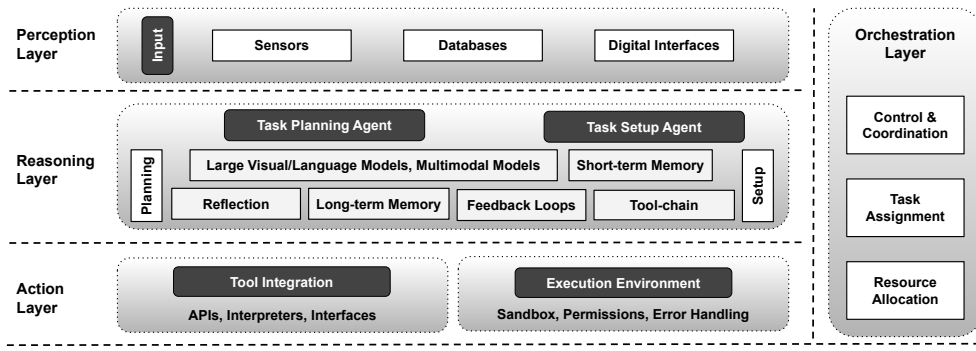


Fig. 1 Illustration of the baseline architecture for an agentic AI system, structured into four distinct layers: *orchestration*, *perception*, *reasoning*, and *action*.

Techniques such as *Reflexion* [50] and *ReAct* [51] integrate online and post-hoc reflective loops, demonstrating substantial gains in robustness and accuracy for task-oriented agents, including reduced hallucinations and improved factual performance. In multi-agent and collaborative settings, reflection mechanisms enable agents to jointly evaluate outcomes, share reasoning traces, and iteratively refine strategies, thereby enhancing collective problem-solving capabilities. Long-term memory facilitates persistent information storage and retrieval across interactions, while feedback loops support ongoing performance evaluation and iterative improvements.

The TSA decomposes goals established by the TPA for a given task into a sequence of actionable steps and selects the appropriate short-term memories and tool chains required to address each query effectively. Short-term memory selection emphasizes the retention and retrieval of relevant contextual information, including prior interactions and intermediate reasoning artifacts such as tool invocations and retrieved data. This memory selection process typically leverages dense vector retrieval mechanisms and layered episodic/semantic memory architectures, enabling the system to maintain awareness of the current task state while also capturing long-term user preferences, thereby enhancing personalization and promoting the reuse of previously acquired insights [52].

The tool-chain selection process involves identification and configuration of the optimal set of tools and functions, such as information retrieval, complex calculations, code execution, or external API integration, to equip the action layer with the necessary capabilities to execute the task effectively and adaptively [4, 6]. The action layer incorporates tool integration and execution environment. Tool integration empowers systems through API connectors for seamless interaction with external services and data sources, code interpreters enabling multi-language execution, document processing for diverse content handling, and database interfaces for robust structured-data management. The execution environment features a sandboxed runtime for secure operation, permission systems to manage tool access, state management for tracking progress, and robust error handling and recovery.

Figure 2 illustrates the states and transitions involved in executing a task within the AgAI system. In the orchestration layer, the Control and Coordination module (CC) interacts with the Task Manager (TAM) and the Resource Manager (RM). The CC is responsible for overall workflow management, including task assignment, progress monitoring, and inter-agent communication. When the TAM schedules a task,

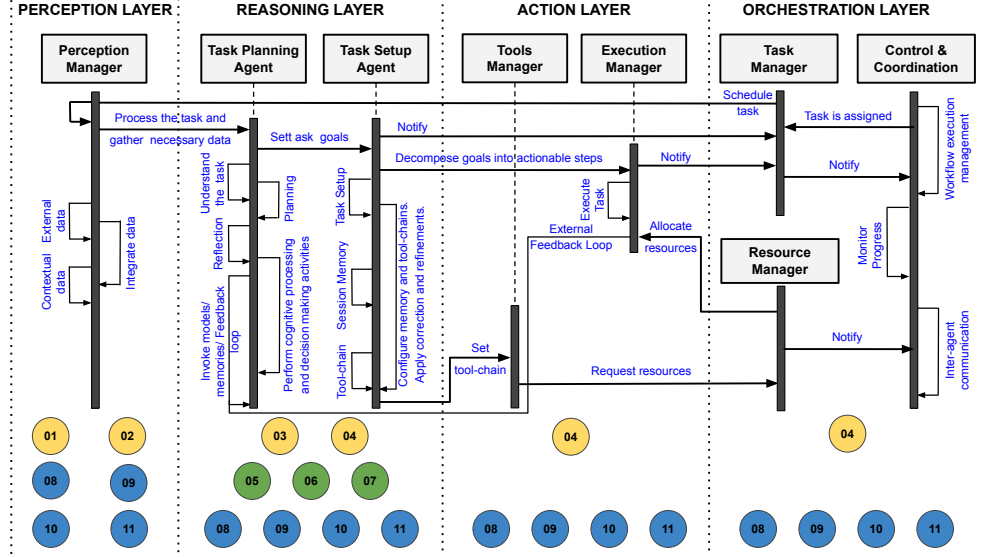


Fig. 2 Illustration of the states and transitions involved in executing a task within the AgAI system, along with the likelihood of the potential attacks at each layer. 01-Evasion, 02-Instruction tuning, 03-Data poisoning, 04-Data exfiltration, 05-Model extraction, 06-Logic corruption, 07-Membership inference, 08-Resource manipulation, 09-DoS, 10-MitM, and 11-Side-channel attacks.

the Perception Manager (PM) processes the task and gathers the necessary contextual and external data, generating the initial input for the reasoning layer. The TPA interprets tasks and coordinates specialized models to perform functions ranging from content generation to visual processing and recommendation systems, while establishing the goals for the assigned task. Supported by robust learning and adaptation mechanisms, the TPA continuously enhances its performance through accumulated experience and feedback from both internal reflections and external execution outcomes. The TSA decomposes goals into actionable steps, configures memory, selects the tool chain, applies corrections and refinements, and notifies the TAM as required by the Tools Manager (TSM) and Execution Manager (EM) in the action layer. The TSM requests resources, such as tools and actuators, from the RM, which in turn allocates the resources to the EM. The EM executes the task based on the information supplied by the SM and sends a completion notification to the TAM. Ultimately, the TAM conveys this information to the CC. Having established the baseline architectural decomposition of AgAI systems and the corresponding task execution states

and transitions, it becomes possible to systematically localize security threats within specific layers and operational flows. This architectural grounding enables a precise mapping of dominant attack vectors to the components where they are most likely to emerge and is sufficiently general to encompass the main classes of AgAI systems.

3.3 Attacks mapping

Table 3 maps primary attack classes (01–11) to their corresponding attack surfaces and to the AgAI layer with the highest likelihood of occurrence. These primary attacks represent foundational threat categories that directly target core components of the perception, reasoning, action, and orchestration layers. They constitute the dominant and most recurrent adversarial strategies observed in both traditional ML systems and emerging agentic AI deployments. Each attack is characterized according to the vulnerabilities and threat vectors associated with specific architectural states and task-execution transitions illustrated in Figure 2. By anchoring attack classes to architectural layers, this mapping clarifies how adversarial actions propagate across interdependent components in multi-agent workflows.

Table 3 Mapping of primary attacks to their corresponding attack surface type and the layer the AgAI architecture with the highest likelihood.

Layer	Attack surface		
	Data	Model	Infrastructure
Perception	01 Evasion attack [14, 25], 02 Instruction-tuning attack [15]	-	08 Resource manipulation [23], 09 Denial-of-service [17], 10 Man-in-the-Middle [24], 11 Side-channel [25]
Reasoning	03 Data poisoning [16, 17], 04 Data exfiltration [21, 22]	05 Model extraction [18], 06 Logic corruption [19, 53], 07 Membership inference [20]	
Action	04 Data exfiltration [21, 22]	-	
Orchestration		-	

Attacks such as evasion [14] and instruction-tuning [15] can broadly target the perception layer. At the reasoning layer, models can be vulnerable to data poisoning [16, 17]. Additionally, data exfiltration [21, 22] can compromise data across the reasoning, action, and orchestration layers. Model extraction [18], logic corruption [19], and membership inference [20] pose significant threats to model security within the reasoning layer. Finally, resource manipulation [23], DoS [17], MitM [24], and side-channel [25] attacks are major concerns that can affect all four layers of AgAI. This layer-wise classification provides three advantages: (i) systematic identification of high-risk components, (ii) improved understanding of cross-layer attack escalation pathways, and (iii) a structured foundation for deriving targeted security and governance controls that preserve CIA properties across dynamic agentic operations.

4 Strategies, and techniques for attacking agentic AI systems

In this section, we discuss the common strategies and techniques targeting the three surfaces, i.e., *data*, *model*, and *infrastructure* for disrupting an AgAI system.

4.1 Data as the attack surface

Recognizing data as the oxygen of AgAI, it is critical to proactively secure the perception layer against threats such as data manipulation and malicious prompt injection to ensure the confidentiality and integrity of the TPA and TSA when executing a given task [54]. Table 4 presents evasion attacks ⑩, which involve the manipulation of input data and can adversely affect the reasoning layer during inference time, consequently compromising all three components of the CIA triad [55]. In evasion attacks, adversaries seek incorrect predictions from a model by introducing minimal modifications (a.k.a perturbations) to the sample targeted for misclassification [18]. These attacks are also known as decision-time attacks. A perturbed text sample, known as an adversarial example, is characterized by subtle modifications, potentially affecting only a few or even one word, thus making it practically imperceptible to human observers [14]. Sponge example generation can be performed using a gradient-based approach that leverages the parameters of a model, or with a genetic algorithm that relies solely on model queries, evolving inputs based on energy or latency metrics [56]. Attacks targeting physical data modification can be carried out by altering sensors or external data [57]. Spoofing techniques involve digital data manipulation using photos, videos, or deepfake.

Table 4 Evasion attacks

Strategies/ techniques	Description	Security property			Impact
		C	I	A	
Adversarial examples	While appearing benign to human evaluators or passing basic validation checks, it is deliberately crafted to induce the AI agent to misclassify, disregard, or mishandle the task.	●	●	●	Bypass detection, execute covert actions
Sponge examples	Maximize neural network neuron activations, inflating computational cost and energy usage.				
Physical data modification	Altering sensors or external data.				
Spoofing	Involve impersonation and deception techniques.				

Table 5 presents instruction-tuning attacks ⑪ that target instruction-tuned LLMs (i.e., models trained and adapted for specific tasks) and represent a class of alignment and capability-adaptation threat that can compromise the CIA properties of the TPA and TSA [15]. These attacks exploit vulnerabilities or limitations introduced during the fine-tuning process using jail-breaking and prompt injection techniques. Jailbreaking of

LLMs refers to the exploitation of vulnerabilities in their alignment training and safety filters to generate outputs for out-of-scope or adversarial queries, thus unlocking functionalities otherwise constrained by ethical and safety considerations. Safety alignment refers specifically to efforts within the reasoning layer to prevent LLMs (or Multi-modal LLMs) from generating harmful content. Weng et al. [58] present a multimodal universal jailbreak approach that integrates image-based and suffix-based attacks. Existing jailbreaking techniques often rely on methods such as applying fuzzing, optimization through search strategies, and even training LLMs specifically to jailbreak other LLMs. Furthermore, there can be prompt-level jailbreaks and token-level jailbreaks [59]. Prompt-level jailbreaks use social engineering and semantic manipulation techniques to bypass language models, while token-level jailbreaks involve adding or adjusting specific tokens to achieve the same effect. Prompt injection attacks intentionally manipulate input data to deceive AgAI into executing a malicious task instead of the intended legitimate one [29].

Table 5 Instruction-tuning

Strategies/ techniques	Description	Security property			Impact
		C	I	A	
Jailbreaking	Attempt to bypass safeguards at runtime or disabling them permanently.	●	●	●	Mis-alignment of LLMs and VLMs
Prompt Injection	Injection of deceptive or hidden instructions within user inputs, documents, or environment data.				

Table 6 presents data poisoning attacks ③ that can compromise the integrity of a TPA and TSA in AgAI, leading to the generation of inaccurate or erroneous outputs for both specific and general tasks [16]. In poisoning attacks, strategies such as injecting,

Table 6 Data poisoning

Strategies/ techniques	Description	Security property			Impact
		C	I	A	
Adversarial examples	The malicious input is designed to permanently compromise the agent’s future decisions by poisoning its learned data or semantic memory.	○	●	○	Corruption, misdirection, or exploitation of training data
Injection, Deletion, Alteration	Use of malicious data or target the datasets and semantic memory.				

deleting, or altering data and adversarial examples are used to influence or corrupt the model learning ability during the training phase, resulting in a defective, biased, or underperforming model [60]. Random label flipping (RLF) attacks, nearest prior label flipping (NPLF) attacks, farthest-prior label flipping (FPLF) attacks, farthest-rotation label flipping (FRLF) attacks, and adversarial label flipping (ALF) attacks are some of the poisoning attacks based on label obscuring pattern [17]. Chen et al. [61] describe

that gradient-based attack methods like the Fast Gradient Sign Method (FGSM) and Basic Iterative Method (BIM) primarily involve propagating noise to the input image using a cost function and subsequently adjusting the input based on the corresponding gradient to craft an adversarial example. Whereas, optimization-based attacks employ well-designed measures such as Limited Memory Broyden-Fletcher-Goldfarb-Shanno (L-BFGS) and Elastic-net Attacks to DNNs (EAD) methods to generate adversarial examples with high confidence and small perturbation.

Table 7 presents data exfiltration ④, defined as the unauthorized extraction or theft of data across the reasoning, action, and orchestration layers of AgAI. Such activities can lead to privacy breaches, intellectual property theft, regulatory non-compliance, and financial loss [62]. Adversaries aim to acquire sensitive information, such as user training data, trained weights, and proprietary algorithms as well as data theft, and data point misclassification. Possible strategies and techniques for data exfiltration involve phishing, malware, privilege escalation, and side-channel [22], thus, essentially impacting the integrity of an AgAI.

Table 7 Data exfiltration

Strategies/ techniques	Description	Security property			Impact
		C	I	A	
Phishing	Malicious payloads embedded within external data or user instructions.	●	○	○	Steal sensitive data from interconnected platforms and tools
Malware	Target APIs, internal tools, or cloud environments.				
Privilege escalation	Exploit trust boundaries, excessive permissions, or compromised credentials.				
Side-channel	Use of malicious payloads or intercept commands, APIs call timing, or memory access behaviors.				

4.2 Model as the attack surface

Attacks targeting models such as LLMs, VLLMs, and MLLMs, which constitute the core of the TPA and TSA in the reasoning layer, present significant security challenges for the safe deployment of AgAI. Adversaries can adopt either a white-box or black-box approach when attacking a model [63]. White-box attacks require access to the model’s internal details, including its architecture, gradients, and training parameters. In contrast, black-box attacks rely solely on analyzing the model’s outputs in response to input queries, without any knowledge of its internal implementation. The threat model for ML primarily includes causative attacks, which involve manipulating the training data, and exploratory attacks, which aim to extract information about the model to facilitate future attacks [17]. Table 8 presents model extraction ⑤, which refers to the act of stealing a model’s architecture, the hyperparameters used during training, its learned parameters, or an approximation of the model’s behaviour [18]. Model extraction is also known as an inference, exploratory, copy, duplication, mimicking, or model approximation attack. Common strategies or techniques for model extraction include

knowledge distillation [53], i.e., a model compression method used to train a smaller version of a pretrained larger network; reverse engineering [64], i.e., learning hardware or software characteristics; and unauthorized access [65], i.e., exploiting cache or timing side channels. Thus, model extraction can affect both confidentiality and integrity.

Table 8 Model extraction

Strategies/ techniques	Description	Security property			Impact
		C	I	A	
Knowledge distillation	Transferring knowledge from a large teacher model to a smaller student model.	●	●	○	Steal proprietary AI models and algorithms used by TPA and TSA
Reverse engineering	Systematically analyzing and inferring the internal structure and parameters of a deployed AI model.				
Unauthorized access	Exploit weak access controls, such as insufficient authentication or unprotected API endpoints.				

Table 9 presents logic corruption 06 that can lead to the manipulation of a model, specifically targeting its parameters or architecture [19]. The primary contributors to logic corruption in a model are data poisoning and backdoor attacks, which affect the model’s integrity [53]. Data poisoning involves injecting malicious data into the training dataset to introduce biases or vulnerabilities in the model that can later be exploited during inference. Backdoor attacks involve inserting deliberately designed

Table 9 Logic corruption

Strategies/ techniques	Description	Security property			Impact
		C	I	A	
Data poisoning	Intentionally injecting malicious or misleading data into training datasets or feedback loops.	○	●	○	Manipulate or subvert TPA and TSA to produce flawed, misleading, biased, or adversarial behaviors
Backdoor	Embed hidden triggers without affecting normal behavior.				
Concept drift	Relationship between input variables and predicted outcomes changes.				
Model decay	Gradual loss of model performance due to changes in input data distributions over time.				

triggers into the training data [66]. This manipulation causes the infected models to exhibit normal behavior when processing benign inputs. However, when the triggers are activated in test data, the performance of the models deteriorates significantly. The accuracy of models (especially in continuous learning systems) can also be adversely affected by concept drift, which refers to unexpected changes in data distribution and the evolution of relationships between attributes and the target feature over time [67]. This typically occurs when real-world conditions shift from those on which the model

was originally trained. In contrast, model decay refers to the deterioration in the performance of an ML model over time, even when the underlying data distribution remains unchanged.

Table 10 presents membership inference attacks ⑩, which aim to determine whether a specific data point was part of a model’s training set [20]. By analyzing model outputs, such as confidence scores, an adversary attempts to infer this membership. In contrast, attribute inference attacks, also known as model inversion attacks, focus on reconstructing sensitive attributes (input features) of the training data by leveraging a model’s outputs or parameters [60]. Similarly, property inference attacks aim to deduce global characteristics of the entire training dataset. These attacks often involve observing how the model’s behavior and prediction loss differ between training and testing data. Fundamentally, these privacy attacks against AgAI exploit the model’s behavior, potentially revealing memorized information about its training data and compromising confidentiality.

Table 10 Membership inference

Strategies/ techniques	Description	Security property			Impact
		C	I	A	
Varying behaviour of the model	variations offer more informative cues for determining whether data was part of the training set.	●	○	○	Determination of data that was used for training TPA and TSA
Prediction loss determination	Smaller loss values indicate training-set data, while larger values suggest otherwise.				

4.3 Infrastructure as the attack surface

Resource manipulation ⑧, DoS ⑨, MitM ⑩, and side-channel attack ⑪ can adversely impact an AgAI across the four layers shown in Figure 1. Adversaries can employ a variety of attack vectors, strategies, and techniques to manipulate resources within these layers. Table 11 presents resource manipulation ⑧, which encompasses tool misuse, workflow disruption, intent breaking, and goal manipulation, all of which can introduce significant and often insidious complexities in AgAI [68]. In particular, tool misuse can impair an agent’s memory and the tool integration process, a vulnerability alarmingly concerning in code generation. Such resource manipulation dramatically elevates the potential for RCE and other code-based attacks. Furthermore, workflow disruption, potentially stemming from embedded backdoors or unauthorized access, can lead to privilege escalation and incorrect decisions throughout an automated workflow [69]. Lastly, prompt injections, compromised data, or malicious tools can subvert intended goals, potentially deceiving human operators or undermining adaptive reasoning and external interaction capabilities [23].

Unauthorized access through identity spoofing, impersonation, or social engineering can provide an initial foothold for compromising critical resources. For example, unauthorized access to the vector database within the TPA and TSA not only exposes stored information but also reveals the structure and relationships between data

Table 11 Resource manipulation

Strategies/ techniques	Description	Security property			Impact
		C	I	A	
Tools misuse, Workflow disruption, Intent-breaking and Goal manipulation	Trigger expensive computations, false or misleading data into communication channels or corrupting coordination, manipulate input or internal states.	●	●	●	Degrade available computational, data, and operational resources
Unauthorized access	Identity spoofing, impersonation, social engineering.				
Contextual data or information	Malicious instructions or misleading context.				
Repudiation and Untraceability	Avoidance of detection, accountability, and attribution.				
Privilege escalation and Isolation	Illegitimate access beyond the original permissions.				
Remote Code Execution (RCE) and code attack	Remote execution of arbitrary code without physical or local access.				
Memory-, and Communication poisoning	Introduction of false, misleading, or malicious data into the memory, inter-agent communication channels or external data sources.				
Sponge attacks	Adversarial inputs crafted to maximize resource usage.				

points [65]. Likewise, unauthorized access to the task execution stack, tool integrations, execution environments, and various resources within the orchestration layer poses significant risks, including the exposure of sensitive data and the potential propagation of malicious code. The manipulation of contextual data at the perception layer can introduce significant security risks, including knowledge poisoning, amplified hallucinations, and indirect prompt injections [70]. These vulnerabilities can, in turn, undermine the decision-making processes and outputs of Retrieval-Augmented Generation (RAG) [16]. RAG serves as a key mechanism that empowers agents with dynamic access to external knowledge within the TPA and TSA. Repudiation and untraceability threats can challenge the integrity of modules such as the task execution stack, tool integrations, and execution environments by impeding log generation and obfuscating the auditability of decision-making processes. Without robust audit and logging of AgAI actions, malicious activity and unusual traffic will likely go undetected.

Privilege escalation and insufficient isolation represent critical vulnerabilities that can be exploited to gain unauthorized access and perpetrate data breaches [71]. Adversaries can exploit privileges to execute RCE in AgAI, enabling the poisoning of code, memory, and communication channels [69]. In a multi-agent environment, the absence of robust privilege isolation can lead to a critical vulnerability. If an agent cannot differentiate between legitimate user requests and maliciously injected instructions, a compromised higher-privilege agent could manipulate other agents into executing unauthorized actions. Memory poisoning attacks can impact both short-term (volatile) and long-term (persistent) memory to achieve malicious objectives such as information corruption, security bypass, and decision manipulation [16]. Short-term memory

poisoning exploits contextual constraints, causing agents to repeat sensitive actions or load tampered data [68]. In contrast, long-term memory poisoning involves injecting persistent false information across sessions, which can corrupt knowledge bases and expose sensitive data. Communication poisoning refers to the manipulation of communication channels between agents to spread false information, disrupt workflows, or influence decision-making ultimately leading to the dissemination of disinformation, reduced operational efficiency, or skewed decision-making processes.

Sponge attacks target either the hardware or the algorithmic complexity of a model [56]. In these attacks, test-time inputs are specifically crafted to increase the energy consumption and prediction latency of hardware accelerators. When applied to large language models (LLMs), sponge attacks can significantly inflate energy usage and latency potentially, by a factor of 10 to 200, indirectly affecting system performance and availability, and potentially leading to DoS attack [72]. Table 12 presents ⑨ denial-of-service (DoS) attacks that tend to exhaust computational resources, causing latency or rendering resources unavailable by overloading AgAI with spurious requests or inputs, thus degrading performance or causing downtime. False positives can also lead to DoS attacks [17]. The main impact of DoS attacks is the unavailability of AgAI. Given the substantial resources the TPA and TSA require, adversary can deliberately craft prompts to diminish model availability [15].

Table 12 Denial-of-service (DoS)

Strategies/ techniques	Description	Security property			Impact
		C	I	A	
Spurious requests	Use of fake, excessive, or random tasks.	○	○	●	Resource exhaustion
False positives	Classify legitimate activity as malicious.				

Table 13 presents man-in-the-middle (MitM) ⑩ that can intercept and manipulate communication channels across various layers of AgAI, posing risks to both confidentiality and integrity. Eavesdropping (intercepting data packets) and data manipulation (tampering with exchanged data) are common techniques in MitM attacks. Wang et

Table 13 Man-in-the-middle (MitM)

Strategies/ techniques	Description	Security property			Impact
		C	I	A	
Eavesdropping	Passively intercepting data.				Intercepts, monitors, or alters information
Data Manipulation	Involves altering or injecting malicious data into the intercepted communication.	●	●	○	

al. [24] discuss the possibility of covertly modifying communication channels between two entities, subsequently deploying malicious payloads to accomplish adversarial objectives. The authors elaborate on the utilization of a generative model, which can be employed to preprocess or enhance incoming raw data for executing MitM attacks

while inferring the model to get a prediction. Mitev et al. [73] demonstrate a MitM attack against personal virtual assistants that hijacks an entire conversation between a voice-controlled virtual assistant and the intended user. The authors demonstrate that programmatic extensions can allow third-party providers to integrate their services with the respective voice assistant to manipulate resources. The attack pipeline involves command jamming, malicious activities invocation, retrieving benign data from benign activities, and transmitting modified data to the user maintaining the interaction semantics from the user’s viewpoint.

Table 14 presents side-channel ⑪ that can exploit both hardware and software components of AgAI to steal model parameters, which can impact the AgAI integrity [18, 64]. For instance, adversaries can deploy queries and keying techniques in the context of machine-learning-as-a-service (MLaaS) to predict the architecture of the target model. Nonfunctional characteristics of a program, such as, computing time, power consumption, electromagnetic radiation, etc., can be the potential techniques to target hardware components for side-channel information leakage [74]. Electromagnetic leakages can result in the extraction of logits for a given input to estimate the model trainable parameters (e.g., weights) or the hyperparameters of the model (e.g., neural network architecture), which can lead to adversarial examples generation, model cloning, and fault injections [75]. Xiang et al. [76] estimate the sparsity of DNN parameters using a side-channel attack to identify the DNN architectures in embedded devices and derive the pre-trained parameter values.

Table 14 Side-channel attacks

Strategies/ techniques	Description	Security property			Impact
		C	I	A	
Queries and keying techniques	Craft queries that exploit responses, resource use, or timing to reveal secret properties or cryptographic keys.	○	●	○	Inference of internal agent states, memory content, actions, or decisions
Electromagnetic leakages	The capture of electromagnetic emissions to infer details regarding the architectures, confidential parameters, or intermediate data processed by LLMs and VLMs.				
Computation time	Exploit variations in computation time for processing different inputs.				
Power consumption	Fine-grained power analysis to deduce model structure, identify specific operations, or extract secret cryptographic keys.				

5 Proposed security and governance framework for agentic AI systems

Existing security frameworks are primarily designed for static ML models or bounded AI applications. Although effective for traditional ML and AI systems, these frameworks are not directly applicable to the strengthening of AgAI security. We adopt

Microsoft framework for securing AI systems [77] and Google Secure AI Framework (SAIF) [78] as foundational baselines (refer A) to derive a framework that can address the security requirements of AgAI systems described using design principles shown in Figure 3.

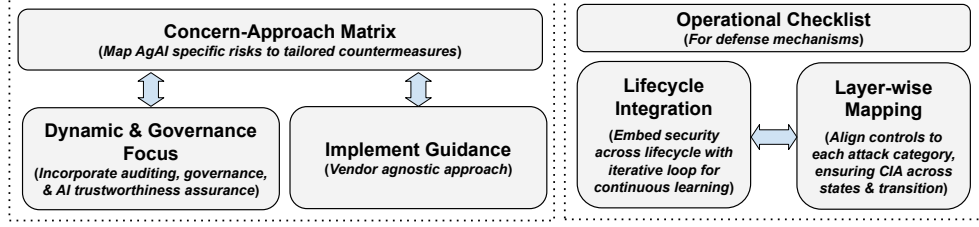


Fig. 3 Illustration of design principles used in deriving a security and governance framework for AgAI systems.

For this purpose, we propose a concern-approach matrix framework to map AgAI-specific risks, limitations, and issues to tailored countermeasures, integrated with dynamic, governance-oriented mechanisms for auditing, traceability, and the assurance of AI trustworthiness, alongside vendor-agnostic strategies to support the effective implementation of guidance. A mapping between life cycle stages and the AgAI layers can be useful in deriving operational checklist for designing effective defense mechanisms. The lifecycle integration embeds security throughout the process, using an iterative loop for continuous learning and the layer-wise mapping aligns controls with each attack category, ensuring CIA compliance across states and transitions to support dynamic workflows.

5.1 Concern-Approach Matrix

It is recommended to identify concerns ranging from minor to critical that are associated with attack surfaces outlined in Tables 3. As shown in Table 15, the identified concerns can be systematically mapped to *security*, *AI/ML risk management*, *privacy and compliance*, and *people and organizational* approaches, culminating the Concern-Approach matrix. This segregation of approaches can enable a structured analysis of corresponding mitigation strategies and tactics for each concern. Security approaches are primarily addressed the design and implementation of defense mechanisms and technical safeguards. AI/ML risk management approaches focus on ensuring robust oversight of models and algorithms throughout their life cycle. Privacy and compliance approaches are directed toward the protection of sensitive data and the fulfillment of regulatory obligations. People and organizational approaches emphasize the role of human factors, institutional processes, and governance structures in sustaining security and resilience.

Table 15: Concern-Approach Matrix

Concerns	Approaches			
	Security	AI/ML Risk Management	Privacy/ Compliance	People/ Organization
Application/Product Security	Tool-chain validation.	Causal reasoning enhancements.	N/A	Cross-functional teams.
Supply Chain Attacks/ Data Lineage/ Retention/ Governance	Secure tool integration.	Data traceability and auditability for feedback loops.	Retention policies for privacy.	Vendor audits.
Autonomy and Inter-Agent Risks	Plan auditing tools.	Formal proofs for multi-agent loops.	Inter-agent consent protocols.	Specialized roles for agent governance.
Resource Integrity and Model Reliability	Adaptive resource allocation to avoid sponge and DoS attacks.	Bias/fairness testing for reasoning.	N/A	Threat modeling workshops. Red teaming exercises.
Model Transparency/ Accountability	Sandboxed execution for actions.	Formal verification of plans/ reflections to counter logic corruption.	Audit logs for membership inference detection.	Governance boards for interpretability reviews.
Data Security/ Sensitive Data Usage	Input validation against evasion/instruction-tuning.	Poisoning detection in memory.	Forgetting mechanisms in long-term memory to prevent exfiltration.	Ethical training on data handling.
Data Poisoning Prevention	Integrity checks in processing. Vector database scanning.	Poisoning detection algorithms in training data.	N/A	Awareness programs on data sourcing risks.
Data Privacy	Encryption across layers (e.g., end-to-end for data flows).	N/A	Data minimization policies.	Privacy officer roles for ongoing oversight.
Dynamic Data Issues	Contextual access controls for real-time/inferred inputs.	Shift detection algorithms for data drift.	Dynamic consent for memory storage/retrieval.	Data provenance and audits.
Access Management	Role-based access with least privilege for layers; end-to-end encryption for MitM.	Query rate-limiting to prevent model extraction.	Differential privacy for data access in perception/reasoning.	Training on roles for multi-agent oversight.
Network/ Endpoint Security	Hardware isolation for side-channel.	Adaptive monitoring for DoS in orchestration.	Secure APIs for tool integration.	Incident response teams.
Threat Detection/ Response	Real-time monitoring.	Incident playbooks for autonomy failures.	Privacy impact assessments.	Response training.
Emerging Regulations/ Governance/Board Reporting	Perimeter controls for infrastructure.	Risk assessments for high-stakes deployments.	Alignment with standards like ISO/IEC 27090, GPDR, HIPAA.	Board-level governance policies.
Error-Prone Manual Reviews/ Anomalies	Automated anomaly detection.	Reflection loops for self-correction.	Automated compliance checks.	AI-assisted review tools with Human supervision.

Human-in-the-Loop (HITL) oversight	Secure interfaces for human intervention in high-risk actions (e.g., approvals in orchestration layer).	Mechanisms for HITL workflows to validate agent decisions and reduce hallucinations.	Consent mechanisms for human review of sensitive data/actions.	Defined roles for human overseers, ethical training on steering AgAI systems above the loop.
Escalation Mechanisms in Multi-Agent Settings	Automated triggers for issue escalation (e.g., coordination failure detection in orchestration).	Algorithms for handoff to human or supervisory agents when limits are reached.	Dynamic privacy checks during escalation to ensure compliance.	Response teams for escalated incidents, governance policies for multi-agent dispute resolution.
Talent Gap	Training related to system security	Training related to AI security.	Compliance workshops on regulations.	AgAI-specific skill development (e.g., autonomy auditing).

Beyond technical vulnerabilities, AgAI also poses significant ethical and socio-technical risks due to its autonomous capabilities [79]. Agentic systems can be weaponized for cyberattacks, disinformation warfare, and large-scale social engineering, while their capacity to process vast multimodal data enables intrusive surveillance and potential authoritarian control. Physical risks emerge through autonomous drones and cyber-physical systems, raising fears of conflict escalation [1, 80]. These risks necessitate interdisciplinary research into ethical governance frameworks, including mandatory transparency in agent decision logs, bias audits tailored to dynamic behaviors, and international standards to prevent dual-use proliferation [81]. Opportunities lie in developing misuse detection mechanisms, such as anomaly-based monitoring for deviant agent actions, and fostering collaborative efforts between technologists, ethicists, and policymakers to align AgAI deployment with societal values, mitigating existential risks while realizing its benefits.

5.2 Operational checklist

Table 16 presents a mapping between the life cycle stages and the AgAI layers, emphasizing end-to-end controls across data handling, model development, deployment, and monitoring. It is recommended to establish an operational checklist that prioritizes defense mechanisms according to the likelihood and potential impact of risks at each stage. Data sources in AgAI systems can be broader than for traditional AI system. In the perception layer, secure data pipelines [82] are needed for handling diverse data sources, including structured and unstructured data, sensor and IoT data, multimodal inputs, user interaction data, simulation and synthetic data, external and contextual data, and internal system data. Similarly, secure storage and processing is required in the perception and reasoning layers. In the reasoning layer, long-term memory protections, using watermarking and backdoor scanning to mitigate model extraction, membership inference, and logic corruption of LLMs and VLMs [83, 84]. Formal proofs can enhance autonomy verification, i.e., agents behaves as intended, reliably, and safely, aligning with integrity requirements for high-stakes reasoning [85].

Table 16: Illustration of the mapping between life cycle stages and the AgAI layers for implementing defense mechanisms.

Life-cycle Stages	AgAI Layers Mapping	Risks		Checklist for defense mechanism
		Likelihood	Impact	
Data collection at design-time and run-time	Perception	High (common input attacks)	High (data compromise leads to downstream errors)	Evasion-resistant validation (e.g., adversarial robustness checks) of data acquired from sensors/interfaces [55]. Add real-time input sanitization to prevent instruction-tuning or perturbed data compromising integrity [40, 58]. Ensure confidentiality via differential privacy for sensitive sources [1, 65].
Data storage and processing	Perception/Reasoning	Medium (supply chain poisoning)	Critical (widespread model corruption)	Secure storage with encryption to block exfiltration; process data using poisoning detection (e.g., anomaly scanning in training sets) [16, 86]. Incorporate feedback loops for dynamic data refinement, with integrity checks against shifts [87].
Model design, training and validation	Reasoning	Medium (extraction/inference attacks)	Critical (IP theft or backdoors)	Use LLMs/VLMs with long-term memory safeguards; apply watermarking/query limits to counter model extraction and membership inference [83, 84]. Validate against logic corruption via backdoor scanning and formal proofs for autonomy [85]. Address integrity breaches from poisoning [16].
Model versioning and deployment	Reasoning/Action	High (deployment DoS/manipulation)	High (operational downtime)	Version models with audit trails for tool-chain selection [88]; deploy in sandboxed environments with permissions to mitigate resource manipulation or DoS [89]. Ensure availability through adaptive allocation for multi-agent deployments [50, 90].
Security testing and monitoring	Action/Orchestration	Medium (ongoing MitM/side-channel)	High (persistent leaks)	Test for MitM/side-channel via end-to-end encryption and hardware isolation; monitor workflows for exfiltration or unintended behaviors [73, 76]. Add reflection auditing to verify plans and inter-agent communications [88].
Incident management and compliance review	All Layers	Low (if proactive)	Critical (regulatory fines/system failure)	Handle incidents with causal analysis for distributional shifts [87]; review compliance using multi-agent governance frameworks [50]. Log all CIA violations for iterative improvements.

In the reasoning and action layers, models versioning, toolchains audits and sandboxed deployment can be used to mitigate resource manipulation and DoS attacks [89]. Adaptive resource allocation can ensure availability in multi-agent setups by dynamically adjusting resource allocation to prevent overloads and maintain operational continuity [90]. MitM [73] and side-channel attacks [76] can be mitigated with run-time tests, such as detecting unauthorized interceptions or covert data leakage within the action and orchestration layers. Exfiltration or unintended behaviors require workflows monitoring to detect over privileged access, insufficient guardrails, and lack of requisite approvals [91]. Audit of reflection mechanism can be useful for validating plans and monitoring inter-agent communication [88]. Incident management and compliance review are strictly required at all layers. Overall, our analysis establishes a structured framework that enables security practitioners and researchers to identify vulnerabilities and devise effective defense mechanisms.

5.3 Concern-Approach matrix and operational checklist in practice

We present the practical application of the Concern-Approach Matrix and the operational checklist using two examples: data poisoning and evasion attacks, as illustrated in Table 17.

Table 17 Application of the Concern-Approach Matrix and the operational checklist

Concern	Layer(s)	Security	AI/ML Risk Management	Privacy/Compliance	People/Organization
Data Poisoning 03	Reasoning	Input validation [16], sandboxing [20]	Bias testing [86], model auditing [92]	GDPR data protection [93]	Red-teaming, training
Evasion Attacks 01	Perception	Sensor data validation [55]	Adversarial training [85]	Consent mechanisms	Simulation testing, training

Example 1. Addressing Data Poisoning (Reasoning Layer)

Data poisoning, which compromises model integrity by injecting malicious data during training or memory updates, is mapped in the Concern-Approach Matrix to the *Reasoning layer* with approaches such as input validation (Security) and bias testing (AI/ML Risk Management).

- *Data Collection and Storage/Processing Stages*: Apply input validation through anomaly scanning and encryption to prevent poisoned data from entering long-term memory or feedback loops, ensuring CIA compliance [55, 65, 94].
- *Model Design, Training, and Validation Stages*: Incorporate bias testing via poisoning detection algorithms [16] and formal proofs for autonomy verification [85], reducing the risk of downstream corruption.
- *Monitoring and Incident Management Stages*: Use reflection auditing and audit trails to monitor for shifts [88], triggering HITL reviews [1, 95] if anomalies are detected.

Example 2. Addressing Evasion Attacks (Perception Layer)

Evasion attacks, involving adversarial inputs that mislead data acquisition (e.g., perturbed sensor data), are aligned in the matrix to the Perception layer with sensor data validation (Security) and simulation testing (People/Organization).

- *Data Acquisition Stage*: Implement evasion-resistant validation, such as adversarial robustness checks [55] and real-time sanitization [40, 58], to safeguard inputs from sensors or interfaces, preserving availability and integrity.
- *Security Testing and Monitoring Stages*: Conduct red-teaming exercises to simulate attacks, identifying vulnerabilities in perception workflows, with end-to-end encryption countering related threats like MitM [96].
- *Incident Management Stage*: Escalate detected evasions via automated triggers and compliance reviews, incorporating people/organization elements like ethical training for overseers to ensure rapid response [93].

6 Conclusions

As attack vectors against autonomous systems become perpetually more sophisticated and agile, the robustness of AgAI systems is emerging as an increasingly critical area of research and development, particularly given the growing demand for AI-driven automation across diverse domains like healthcare, finance, and transportation. This paper advances the field by proposing a reference architecture of an AgAI system that consists of four layers: *perception*, *reasoning*, *action*, and *orchestration*. We determine the states and transitions involved in task execution to delineate security requirements for the AgAI system ensuring CIA properties.

We systematically categorize attacks across data, model, and infrastructure surfaces including evasion, poisoning, model extraction, and resource manipulation and subsequently examine potential attack vectors, strategies, and techniques that can be used to exploit each surface within an AgAI system. Our analysis provides essential insights into the nature and extent of threats in AgAI systems. The Concern-Approach matrix provides a solid foundation for the development of stringent security and governance frameworks that can regulate the deployment and operation of AgAI technologies. Furthermore, the mapping between life cycle stages and AgAI layers serves as a basis for deriving an operational checklist to guide the implementation of defense mechanisms.

Declaration

Data availability statement

Data sharing not applicable to this article as no datasets were generated or analysed during the current study.

Consent to publish

Not applicable

Consent to participate

Not applicable. This study does not involve human participants, animals, or clinical trials.

Ethical Approval and accordance

Not application

Clinical trial number

Not applicable

Funding

Funding information will be provided upon acceptance of the manuscript.

References

- [1] Acharya, D.B., Kuppan, K., Divya, B.: Agentic ai: Autonomous intelligence for complex goals-a comprehensive survey. *IEEE Access* (2025)
- [2] Ramya, J., Yerraguravagari, S.S., Gaikwad, S., Gupta, R.K., et al.: Ai and machine learning in predictive analytics: Revolutionizing business strategies through big data insights. *Library of Progress-Library Science, Information Technology & Computer* **44**(3) (2024)
- [3] Narayanaswamy, A.: Evolution of microsoft copilot. In: *Proceedings of the Microsoft Copilot for Windows 11: Understanding the AI-Powered Features in Windows 11*, pp. 1–24. Springer, ??? (2024)
- [4] Google: Safety and security for ai agents. Technical report, Google (2026). Accessed: 01-05-2026
- [5] Anthropic: Make ai agents your unfair advantage. Technical report, Anthropic (2026). Accessed: 01-05-2026
- [6] Microsoft: Ai agents are changing the way we work. Technical report, Microsoft (2026). Accessed: 01-05-2026
- [7] OpenAI: Operator. Technical report, OpenAI (2026). Accessed: 01-05-2026
- [8] IBM: Ai agents that make your job easier. Technical report, IBM (2026). Accessed: 01-05-2026
- [9] Raptis, E.K., Kapoutsis, A.C., Kosmatopoulos, E.B.: Agentic llm-based robotic systems for real-world applications: a review on their agenticness and ethics. *Frontiers in Robotics and AI* **12**, 1605405 (2025)
- [10] Rainer Jr, R.K., Richey Jr, R.G., Chowdhury, S.: How robotics is shaping digital logistics and supply chain management: An ongoing call for research. *Journal of Business Logistics* **46**(1), 70005 (2025)
- [11] Bastola, A., Wang, H., Boroujeni, S.P.H., Brinkley, J., Moshayedi, A.J., Razi, A.: Driving towards inclusion: A systematic review of ai-powered accessibility enhancements for people with disability in autonomous vehicles. *IEEE Access* (2025)
- [12] Bayer: A growing healthcare demand and new technologies are transforming the pharmaceutical industry. Technical report, Bayer (2026). Accessed: 01-05-2026
- [13] Gartner: Gartner unveils top predictions for it organizations and users in 2025 and beyond. Technical report, Gartner (2026). Accessed: 01-05-2026
- [14] Vitorino, J., Maia, E., Praça, I.: Adversarial evasion attack efficiency against large

- language models. In: Proceedings of the International Symposium on Distributed Computing and Artificial Intelligence, pp. 14–22 (2024). Springer
- [15] Yao, Y., Duan, J., Xu, K., Cai, Y., Sun, Z., Zhang, Y.: A survey on large language model (llm) security and privacy: The good, the bad, and the ugly. *High-Confidence Computing*, 100211 (2024)
 - [16] Chen, Z., Xiang, Z., Xiao, C., Song, D., Li, B.: Agentpoison: Red-teaming llm agents via poisoning memory or knowledge bases. *Advances in Neural Information Processing Systems* **37**, 130185–130213 (2024)
 - [17] Wang, X., Li, J., Kuang, X., Tan, Y.-a., Li, J.: The security of machine learning in an adversarial setting: A survey. *Journal of Parallel and Distributed Computing* **130**, 12–23 (2019)
 - [18] Oliynyk, D., Mayer, R., Rauber, A.: I know what you trained last summer: A survey on stealing machine learning models and defences. *ACM Computing Surveys* (2023)
 - [19] Jeong, D.: Artificial intelligence security threat, crime, and forensics: Taxonomy and open issues. *IEEE Access* **8**, 184560–184574 (2020)
 - [20] Hu, H., Salcic, Z., Sun, L., Dobbie, G., Yu, P.S., Zhang, X.: Membership inference attacks on machine learning: A survey. *ACM Computing Surveys (CSUR)* **54**(11s), 1–37 (2022)
 - [21] Gupta, S., Crispo, B.: Towards autonomous device protection using behavioural profiling and generative artificial intelligence. *IET Cyber-Physical Systems: Theory & Applications* **10**(1), 12102 (2025)
 - [22] Sabir, B., Ullah, F., Babar, M.A., Gaire, R.: Machine learning for detecting data exfiltration: A review. *ACM Computing Surveys (CSUR)* **54**(3), 1–47 (2021)
 - [23] Watson, N., Hessami, A., Fassihi, F., Abbasi, S., Jahankhani, H., El-Deeb, S., Caetano, I., David, S., Newman, M., Moriarty, S.: Guidelines for agentic ai safety volume 1: Agentic ai safety experts focus group-sept. Technical report, Agentic AI Safety Experts Focus Group (2026). Accessed: 01-05-2026
 - [24] Wang, D., Li, C., Wen, S., Nepal, S., Xiang, Y.: Man-in-the-middle attacks against machine learning classifiers via malicious generative models. *IEEE Transactions on Dependable and Secure Computing* **18**(5), 2074–2087 (2020)
 - [25] Gupta, S., Maple, C., Passerone, R.: An investigation of cyber-attacks and security mechanisms for connected and autonomous vehicles. *IEEE Access* (2023)
 - [26] He, F., Zhu, T., Ye, D., Liu, B., Zhou, W., Yu, P.S.: The emerged security and privacy of LLM agent: A survey with case studies. *ACM Comput. Surv.* **58**(6),

162–116236 (2026) <https://doi.org/10.1145/3773080>

- [27] Tang, Y., Liu, Y., Lan, J., Yan, Z., Gelenbe, E.: Security of llm-based agents regarding attacks, defenses, and applications: A comprehensive survey. *Information Fusion* **127**, 103941 (2026) <https://doi.org/10.1016/j.inffus.2025.103941>
- [28] Ferrag, M.A., Tihanyi, N., Hamouda, D., Maglaras, L., Lakas, A., Debbah, M.: From prompt injections to protocol exploits: Threats in llm-powered ai agents workflows. *ICT Express* (2025) <https://doi.org/10.1016/j.ict.2025.12.001>
- [29] Liu, Y., Jia, Y., Geng, R., Jia, J., Gong, N.Z.: Formalizing and benchmarking prompt injection attacks and defenses. In: *Proceedings of the 33rd USENIX Security Symposium (USENIX Security 24)*, pp. 1831–1847 (2024)
- [30] Bombieri, M., Ponzetto, S.P., Rospocher, M.: The dangerous effects of a frustratingly easy llms jailbreak attack. *IEEE Access* **13**, 126418–126431 (2025) <https://doi.org/10.1109/ACCESS.2025.3589112>
- [31] Zhan, Q., Liang, Z., Ying, Z., Kang, D.: InjecAgent: Benchmarking indirect prompt injections in tool-integrated large language model agents. In: Ku, L.-W., Martins, A., Srikumar, V. (eds.) *Findings of the Association for Computational Linguistics: ACL 2024*, pp. 10471–10506. Association for Computational Linguistics, Bangkok, Thailand (2024). <https://doi.org/10.18653/v1/2024.findings-acl.624> . <https://aclanthology.org/2024.findings-acl.624/>
- [32] Zou, W., Geng, R., Wang, B., Jia, J.: Poisonedrag: Knowledge corruption attacks to retrieval-augmented generation of large language models. In: Bauer, L., Pellegrino, G. (eds.) *34th USENIX Security Symposium, USENIX Security 2025, Seattle, WA, USA, August 13-15, 2025*, pp. 3827–3844. USENIX Association, ??? (2025). <https://www.usenix.org/conference/usenixsecurity25/presentation/zou-poisonedrag>
- [33] Zhang, H., Huang, J., Mei, K., Yao, Y., Wang, Z., Zhan, C., Wang, H., Zhang, Y.: Agent security bench (ASB): formalizing and benchmarking attacks and defenses in llm-based agents. In: *The Thirteenth International Conference on Learning Representations, ICLR 2025, Singapore, April 24-28, 2025*. OpenReview.net, ??? (2025). <https://openreview.net/forum?id=V4y0CpX4hK>
- [34] Sapkota, R., Roumeliotis, K.I., Karkee, M.: Ai agents vs. agentic ai: A conceptual taxonomy, applications and challenges. *Information Fusion* **126**, 103599 (2026)
- [35] Kshetri, N.: From predictive and generative to agentic ai: Shaping the future of marketing operations and strategies. *Computer* **58**(4), 121–129 (2025)
- [36] Adversa: Top ai security incidents. Technical report, Adversa (2026). Accessed: 01-05-2026

- [37] Anthropic: Threat intelligence report: August 2025. Technical report, Anthropic (2026). Accessed: 01-05-2026
- [38] NIST: Cve-2025-54794. Technical report, NIST (2026). Accessed: 01-05-2026
- [39] Zhu, Y., Miao, C., Xue, H., Yu, Y., Su, L., Qiao, C.: Malicious attacks against multi-sensor fusion in autonomous driving. In: Proceedings of the 30th Annual International Conference on Mobile Computing and Networking, pp. 436–451 (2024)
- [40] Che, X., Gao, P., Zhu, M., Gu, H., Li, Q., Li, M.: Text-enhanced method for llms domain adaptation in cybersecurity. *Journal of Computer Information Systems*, 1–16 (2025)
- [41] Saraf, I., Balassiano, O.: Model namespace reuse: An ai supply-chain attack exploiting model name trust. Technical report, Palo Alto Networks (2026). Accessed: 01-05-2026
- [42] Court, U.: The usd 7 million betrayal: Inside the xai-openai trade secret theft case. Technical report, Breached (2026). Accessed: 01-05-2026
- [43] Rubin, S.: Unit 42 develops agentic ai attack framework. Technical report, Palo Alto Networks (2026). Accessed: 01-05-2026
- [44] IBM: Ibm x-force 2025 threat intelligence index. Technical report, IBM (2026). Accessed: 01-05-2026
- [45] Lakshmanan, R.: Salesforce patches critical forcedleak bug exposing crm data via ai prompt injection. Technical report, The hacker news (2026). Accessed: 01-05-2026
- [46] Alder, S.: Unsecured serviceaide database exposed data of 483,000 catholic health patients. Technical report, HIPAA (2026). Accessed: 01-05-2026
- [47] Baran, G.: Hackers exploit copilot ai for sharepoint to access passwords & other sensitive data. Technical report, Cyber Security News (2026). Accessed: 01-05-2026
- [48] Gill, J.K.: Ai agent infrastructure stack for agentic systems. Technical report, Xenonstack (2026). Accessed: 01-05-2026
- [49] Burden, D., Savin-Baden, M.: Virtual humans: Today and tomorrow. In: Proceedings of the Virtual Humans: Today and Tomorrow. Chapman and Hall/CRC, ??? (2019)
- [50] Bo, X., Zhang, Z., Dai, Q., Feng, X., Wang, L., Li, R., Chen, X., Wen, J.-R.: Reflective multi-agent collaboration based on large language models. *Advances in Neural Information Processing Systems* **37**, 138595–138631 (2024)

- [51] Yao, S., Zhao, J., Yu, D., Du, N., Shafran, I., Narasimhan, K.R., Cao, Y.: React: Synergizing reasoning and acting in language models. In: Proceedings of the Eleventh International Conference on Learning Representations (2022)
- [52] De Brigard, F., Umanath, S., Irish, M.: Rethinking the distinction between episodic and semantic memory: Insights from the past, present, and future. *Memory & Cognition* **50**(3), 459–463 (2022)
- [53] Wang, C., Chen, J., Yang, Y., Ma, X., Liu, J.: Poisoning attacks and countermeasures in intelligent networks: Status quo and prospects. *Digital Communications and Networks* **8**(2), 225–234 (2022)
- [54] Liang, W., Tadesse, G.A., Ho, D., Fei-Fei, L., Zaharia, M., Zhang, C., Zou, J.: Advances, challenges and opportunities in creating data for trustworthy ai. *Nature Machine Intelligence* **4**(8), 669–677 (2022)
- [55] Gupta, S., Raja, K., Passerone, R.: Visual prompt engineering for enhancing facial recognition systems robustness against evasion attacks. *IEEE Access* (2024)
- [56] Shumailov, I., Zhao, Y., Bates, D., Papernot, N., Mullins, R., Anderson, R.: Sponge examples: Energy-latency attacks on neural networks. In: Proceedings of the IEEE European Symposium on Security and Privacy (EuroS&P), pp. 212–231 (2021). IEEE
- [57] Kurakin, A., Goodfellow, I.J., Bengio, S.: Adversarial examples in the physical world. In: *Artificial Intelligence Safety and Security*, pp. 99–112. Chapman and Hall/CRC, ??? (2018)
- [58] Wang, Y., Hu, W., Dong, Y., Liu, J., Zhang, H., Hong, R.: Align is not enough: Multimodal universal jailbreak attack against multimodal large language models. *IEEE Transactions on Circuits and Systems for Video Technology* (2025)
- [59] Rababah, B., Wu, S.T., Kwiatkowski, M., Leung, C.K., Akcora, C.G.: Are existing large language models robust against jailbreak attacks? In: Proceedings of the IEEE International Conference on Big Data (BigData), pp. 5383–5391 (2024). IEEE
- [60] Zhou, S., Liu, C., Ye, D., Zhu, T., Zhou, W., Yu, P.S.: Adversarial attacks and defenses in deep learning: From a perspective of cybersecurity. *ACM Computing Surveys* **55**(8), 1–39 (2022)
- [61] Chen, Y., Zhang, M., Li, J., Kuang, X.: Adversarial attacks and defenses in image classification: A practical perspective. In: Proceedings of the 7th International Conference on Image, Vision and Computing (ICIVC), pp. 424–430 (2022). IEEE
- [62] Chung, M.-H., Chignell, M., Wang, L., Jovicic, A., Raman, A.: Interactive machine learning for data exfiltration detection: Active learning with human

- expertise. In: Proceedings of the International Conference on Systems, Man, and Cybernetics (SMC), pp. 280–287 (2020). IEEE
- [63] Yuan, X., He, P., Zhu, Q., Li, X.: Adversarial examples: Attacks and defenses for deep learning. *IEEE transactions on neural networks and learning systems* **30**(9), 2805–2824 (2019)
 - [64] Hoseini, S., Suutala, J., Partala, J., Halunen, K.: Threat modeling ai/ml with the attack tree. *IEEE Access* (2024)
 - [65] Huang, K., Huang, G., Dawson, A., Wu, D.: Genai application level security. In: Proceedings of the Generative AI Security: Theories and Practices, pp. 199–237. Springer, ??? (2024)
 - [66] Han, S., Lin, C., Shen, C., Wang, Q., Guan, X.: Interpreting adversarial examples in deep learning: A review. *ACM Computing Surveys* (2023)
 - [67] Suárez-Cetrulo, A.L., Quintana, D., Cervantes, A.: A survey on machine learning for recurring concept drifting data streams. *Expert Systems with Applications* **213**, 118934 (2023)
 - [68] OWASP: Owasp ai security and privacy guide. Technical report, OWASP (2026). Accessed: 01-05-2026
 - [69] Deng, Z., Guo, Y., Han, C., Ma, W., Xiong, J., Wen, S., Xiang, Y.: Ai agents under threat: A survey of key security challenges and future pathways. *ACM Computing Surveys* **57**(7), 1–36 (2025)
 - [70] Huang, L., Yu, W., Ma, W., Zhong, W., Feng, Z., Wang, H., Chen, Q., Peng, W., Feng, X., Qin, B., *et al.*: A survey on hallucination in large language models: Principles, taxonomy, challenges, and open questions. *ACM Transactions on Information Systems* **43**(2), 1–55 (2025)
 - [71] Ruii, D.: Llms red teaming. In: Large Language Models in Cybersecurity: Threats, Exposure and Mitigation, pp. 213–223. Springer, ??? (2024)
 - [72] Cinà, A.E., Demontis, A., Biggio, B., Roli, F., Pelillo, M.: Energy-latency attacks via sponge poisoning. *Information Sciences* **702**, 121905 (2025)
 - [73] Mitev, R., Miettinen, M., Sadeghi, A.-R.: Alexa lied to me: Skill-based man-in-the-middle attacks on virtual assistants. In: Proceedings of the 2019 ACM Asia Conference on Computer and Communications Security, pp. 465–478 (2019)
 - [74] Păsăreanu, C.S., Kersten, R., Luckow, K., Phan, Q.-S.: Symbolic execution and recent applications to worst-case execution, load testing, and security analysis. *Advances in Computers* **113**, 289–314 (2019)
 - [75] Coqueret, B., Carbone, M., Sentieys, O., Zaid, G.: When side-channel attacks

- break the black-box property of embedded artificial intelligence. In: Proceedings of the 16th ACM Workshop on Artificial Intelligence and Security, pp. 127–138 (2023)
- [76] Xiang, Y., Chen, Z., Chen, Z., Fang, Z., Hao, H., Chen, J., Liu, Y., Wu, Z., Xuan, Q., Yang, X.: Open dnn box by power side-channel attack. *IEEE Transactions on Circuits and Systems II: Express Briefs* **67**(11), 2717–2721 (2020)
 - [77] Microsoft: Best practices for ai security risk management. Technical report, Microsoft (2026). Accessed: 01-05-2026
 - [78] Google: Google’s secure ai framework. Technical report, Google (2026). Accessed: 01-05-2026
 - [79] Hosseini, S., Seilani, H.: The role of agentic ai in shaping a smart future: A systematic review. *Array*, 100399 (2025)
 - [80] Schleiger, E., Mason, C., Naughtin, C., Reeson, A., Paris, C.: Collaborative intelligence: A scoping review of current applications. *Applied Artificial Intelligence* **38**(1), 2327890 (2024)
 - [81] Murugesan, S.: The rise of agentic ai: Implications, concerns, and the path forward. *IEEE Intelligent Systems* **40**(2), 8–14 (2025)
 - [82] Seshadri, K., Chandran, R., Chandrashekar, S., CURTIS, T.J., ASAWA, A., Devarajan, R.: Methods and apparatus for data pipelines between cloud computing platforms. Technical report (May 31 2022). US Patent 11,347,572
 - [83] Wan, S., Liu, W., Liu, Y., Yuan, F., Meng, C.: Watermarking vision-language models. In: Proceedings of the 6th ACM International Conference on Multimedia in Asia, pp. 1–1 (2024)
 - [84] Wang, Z., Mu, C., Hu, S., Chu, C., Li, X.: Membership inference via backdooring. In: Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence (IJCAI-22). IJCAI Organization, ??? (2022)
 - [85] Gupta, S., Passerone, R.: An investigation of visual foundation models robustness. *Machine Learning* (2025)
 - [86] Wang, Z., Ma, J., Wang, X., Hu, J., Qin, Z., Ren, K.: Threats to training: A survey of poisoning attacks and defenses on machine learning systems. *ACM Computing Surveys* **55**(7), 1–36 (2022)
 - [87] Thulasidasan, S., Thapa, S., Dhaubhadel, S., Chennupati, G., Bhattacharya, T., Bilmes, J.: An effective baseline for robustness to distributional shift. In: Proceedings of the 20th IEEE International Conference on Machine Learning and Applications (ICMLA), pp. 278–285 (2021). IEEE

- [88] Schreyer, M., Gu, H., Moffitt, K., Vasarhelyi, M.A.: Artificial intelligence agentic auditing. SSRN (2024)
- [89] Shavit, Y., Agarwal, S., Brundage, M., Adler, S., O’Keefe, C., Campbell, R., Lee, T., Mishkin, P., Eloundou, T., Hickey, A., et al.: Practices for governing agentic ai systems. OpenAI Research Paper (2023)
- [90] Hady, M.A., Hu, S., Pratama, M., Cao, Z., Kowalczyk, R.: Multi-agent reinforcement learning for resources allocation optimization: a survey. *Artificial Intelligence Review* **58**(11), 354 (2025)
- [91] Boskabadi, M.R., Cao, Y., Khadem, B., Clements, W., Gerek, Z.N., Reuthe, E., Sivaram, A., Savoie, C.J., Mansouri, S.S.: Industrial agentic ai and generative modeling in complex systems. *Current Opinion in Chemical Engineering* **48**, 101150 (2025)
- [92] Lin, C.-J.: Building resilient ai models against data poisoning attacks. *Multidisciplinary Studies and Innovations* **3**(4), 1–16 (2022)
- [93] Adomaitis, L., Oak, R.: Ethics of adversarial machine learning and data poisoning. *Digital Society* **2**(1), 8 (2023)
- [94] Zhao, P., Jiang, H., Li, J., Xiao, Z., Liu, D., Ren, J., Guo, D.: Garbage in, garbage out: Poisoning attacks disguised with plausible mobility in data aggregation. *IEEE Transactions on Network Science and Engineering* **8**(3), 2679–2693 (2021)
- [95] Krishnan, N.: Ai agents: Evolution, architecture, and real-world applications. arXiv preprint arXiv:2503.12687 (2025)
- [96] Wang, S., Ko, R.K., Bai, G., Dong, N., Choi, T., Zhang, Y.: Evasion attack and defense on machine learning models in cyber-physical systems: A survey. *IEEE communications surveys & tutorials* **26**(2), 930–966 (2023)

A Security frameworks

A.1 Microsoft

The framework examines both intentional and unintentional failures that can compromise the security of ML systems. Intentionally motivated failures include perturbation attacks, data poisoning, model inversion, membership inference, model stealing, ML system reprogramming, physical adversarial examples, malicious ML providers, supply chain attacks, and backdoor entries. Unintentional failures can arise from reward hacking, side effects, distributional shifts, natural adversarial examples, common corruptions, or incomplete testing.

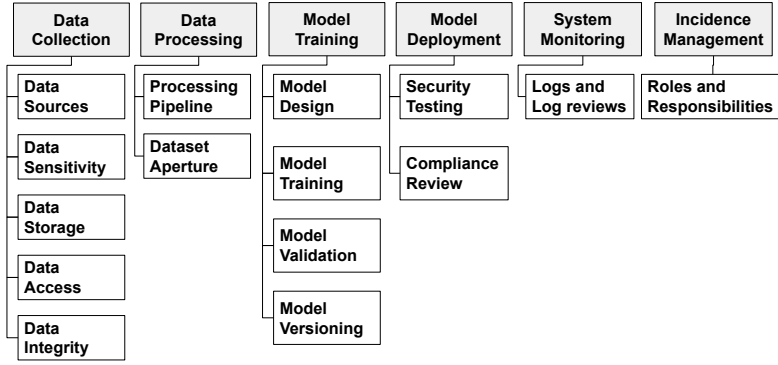


Fig. 4 Illustration of technical controls to secure AI systems throughout their lifecycle [77].

A.2 Google

The SAIF framework recommends strengthening security foundations within the AI ecosystem and integrating AI into an organization’s threat model to enhance detection and response capabilities. Further, it suggests automating defenses to effectively counter both existing and emerging threats and standardize platform-level controls to ensure consistent security standards across the organization. Table 5 outlines approaches for addressing the primary concerns of security and risk professionals.

Security	AI/ML model risk management	Privacy and compliance	People and organization
Access management	Model transparency and accountability	Data privacy and usage of sensitive data	Talent gap
Network / endpoint security	Error-prone manual reviews for detecting anomalies	Emerging regulations	Governance / Board reporting
Application / product security	Data poisoning prevention		
Supply chain attacks	Data lineage, retention and governance controls		
Data security			
AI specific threats			
Threat detection and response			

Fig. 5 SAIF: primary concerns vs. approaches to address the security and risk in AI systems [78]

Biographies



Sandeep Gupta received the Ph.D. degree in information and communication technology from the University of Trento, Italy. He is currently a Principal Engineer with Queen's University Belfast, U.K. He worked with Samsung, Accenture, and Mentor Graphics in information technology domains driving several research projects and products from their incubation to next-generation iteration. He was a recipient of the prestigious Marie Skłodowska-Curie Research Fellowship. He has published more than 40 papers in peer-reviewed journals and leading conferences. His research interests include AI for security, security for AI systems, biometric-based identity and access mechanisms, and usable security and privacy solutions for cyber-physical systems and the IoT.



Marco Bombieri received the Ph.D. degree in computer science from the University of Verona, Italy, working within the ERC Project ARS (Autonomous Robotic Surgery). He is currently an Assistant Professor at the Department of Information Engineering and Computer Science of the University of Trento, Italy. His research interests include agentic artificial intelligence, natural language processing, and large language models with applications in the medical domain.



Bruno Crispo received his Ph.D. degree in computer science from the University of Cambridge, UK, in 1999, having received the M.Sc. degree in computer science from the University of Turin, Italy, in 1993. He is professor at the University of Trento since September 2005. Before that, he was an associate professor at Vrije Universiteit in Amsterdam and full professor at KULeuven in Belgium. His main interests span the field of security and privacy. In particular, his recent work focuses on IoT security, system and network security, web security, behavioural biometric, and access control. He has published more than 220 papers in international journals and conferences on security-related topics.



Paolo Giorgini is a Full Professor in the Department of Information Engineering and Computer Science at the University of Trento, Italy, where he previously served as Department Director. His research interests lie at the intersection of Artificial Intelligence and Software Engineering, with a particular focus on autonomous and evolving software agents, goal-oriented and agent-oriented software engineering, and security requirements modeling. He has made significant contributions to the development of requirements modeling languages, as well as methodologies and tools for software and security engineering. In 2015, he received the 10-Year Most Influential Paper Award at the International Conference on Requirements Engineering. His publication record includes over 300 refereed journal and conference papers.