AI and Machine Learning Advances
Volume 2, Issue 2
ISSN: 3067-3216

Unified Security Orchestration Framework for Protecting AI/ML Systems Against Input-Based and Model Supply Chain Attacks

[1]*Senthil Muthu
[1]Independent Researcher.

Abstract

The quick implementation of Large Language Models (LLMs) in companies and vital systems has exposed new types of security concerns that are not solvable by current cybersecurity tools. These tools are now vulnerable to adversarial attacks like prompt injection, indirect manipulation of input, multi-turn exploits and compromises of the supply chains of models. The current security systems are distributed, protecting specific parts of AI so they do not cover the risks of coordinated or new attacks. The paper presents a new Unified Security Orchestration Framework (USOF), a security control system independent from models that provides security management across all stages of AI/ML. The framework comprises six tightly integrated modules: an Input Trust and Threat Analysis Module (ITTAM) to identify input-based attacks across direct, indirect, multi-turn, and multimodal vectors; a Context-Aware Policy Enforcement Engine (CAPEE) to enforce dynamic, context-sensitive policies at runtime; an Execution Isolation Layer (EIL) to establish safe operational boundaries for agentic AI activities; a Model Integrity and Supply Chain Validator (MISCV) to detect supply chain compromise through training-data-independent behavioral fingerprinting; a Response Governance Engine (RGE) to screen model outputs for sensitive data leakage and second-order injection; and an Adaptive Learning and Feedback Mechanism (ALFM) to evolve detection capability from confirmed threat events continuously. The proposed structure provides a unified orchestration strategy in which threats are identified, policies enforced, and responses coordinated across all stages of the AI system lifecycle. It supports agent-based and multi-model architectures, single-model, and is supported on clouds, on-premises, and on the edge. USOF is an effective and scalable approach to the security of modern AI deployed in high-risk settings (finance, healthcare, etc.) by filling the input-level and model-level vulnerabilities of a single integrated system and critical infrastructure.

Keywords: AI Security, Prompt Injection, Model Supply Chain, Behavioral Fingerprinting, Security Orchestration, LLM Security, Policy Enforcement, Execution Isolation, Adversarial Machine Learning.

1 Introduction

Enterprise software architectures have been fundamentally changed with the introduction of Large Language Models (LLMs) and the larger category of machine learning systems. Since code generation assists and document analysis pipes to multi-agent autonomous workflows, AI systems have now been integrated into the decision-critical processes of finance, healthcare, critical infrastructure, and enterprise operations. This is a fast-paced adoption at the expense of the creation of corresponding security structures, which has led to a perilous imbalance between capability and protection. The conventional cybersecurity models that are based on network boundaries, access control lists, and signature-based detection of threats are structurally insufficient to address the new attack surfaces posed by AI systems. Prompt injection was formally included in the OWASP Top 10 of Large Language Model Applications, which was first published in 2023; the natural language interface between users and artificial intelligence systems has created fundamentally new attack vectors that no traditional security control measure was ever intended to counter (Derner et al., 2024). With well-designed language inputs, attackers can control behavior by manipulating models, bypass access controls without having to exploit code vulnerabilities, exfiltrate sensitive data without activating traditional data loss prevention systems or compromise AI systems at the model supply chain level in a way that is completely undetected by runtime monitoring tools.

The current literature on security research has yielded

Senthil Muthu
Independent Researcher
Email: muthu.senthil@gmail.com

Received: 24-Mar-2026 Revised: 9-Apr-2026 Accepted: 13-May-2026

AI and Machine Learning Advances
Volume 2, Issue 2
ISSN: 3067-3216

considerable information on the categories of threats that apply at the individual level. The scale of indirect prompt injection attacks was shown by (Yi et al., 2025), where adversarial instructions that are found in documents loaded by AI systems during runtime can take control of model behavior, without any direct interaction between the attacker and the system. They listed the exploitability of prompt injection methods in practice in the commercial applications of LLM integration. (Gu et al., 2017) included the principal conceptualisation of BadNets - backdoored neural networks - to establish that machine learning supply chains have a fundamental vulnerability to adversarial interference during model training and distribution. applied this to contemporary pre-trained model ecosystems, demonstrating that supply chain poisoning attacks are resistant to high-fidelity downstream fine-tuning procedures.

Despite this literature, no deployed security solution offers end-to-end protection to the entire lifecycle of the AI pipeline. Current solutions are point solutions: classifiers detecting separate malicious prompts, output filters redacting particular data categories, or model evaluation systems that can check integrity at deployment time. None of them are integrated, real-time, adaptive security orchestration layers that have the ability to simultaneously regulate inputs, policies, execution environments, model integrity and outputs and continuously learn as a result of detected threats.

This research introduces the Unified Security Orchestration Framework (USOF). This architectural model fills the gap between artificial intelligence and machine learning (AI and ML) capabilities and enterprise security requirements. The USOF provides an integrated security control plane for data flows in AI systems and operates orthogonally relative to any specific AI model architecture/deployment environment across six interrelated modules. The major contributions of this research include: (1) The development of a comprehensive architecture to perform unified orchestration of AI and ML security; (2) New techniques to detect multiple types of prompt injection attacks using multi-turn stateful tracking and retrieval-aware provenance of the Recurrent Actor–Agent (RAG) systems; and (3) Model integrity checking methodology which does not require access to training data, in order to detect compromise through behaviours fingerprinting and statistical drift analysis.

The remainder of this paper is structured as follows: In Section II, we characterise the threat landscape and articulate the research problem. In Section III, we present a survey of related works and identify gaps in knowledge and research. In Section IV, we describe the architecture of the USOF system. In Section V, we provide a detailed description of each of the six modules. In Section VI, we describe use cases for the application of the USOF across representative environments and deployment scenarios. In Section VII, we compare USOF to current methods, describe the strengths and weaknesses of the USOF, and present future research directions. In Section VIII, we summarise and conclude the paper.

2 Threat Landscape and Problem Formulation

This section characterises the threat landscape facing modern AI/ML systems, organising threats into two primary categories: input-layer attacks and model supply chain attacks. Together, these categories define the full attack surface that a unified security framework must address.

2.1 Input-Layer Attacks

The natural language interface between users and AI systems is used to pose input-layer attacks, which make a model behave in a way that avoids the desired security controls. These attacks are especially difficult since the same natural language interface that allows AI utility also defines the attack surface, and therefore, it is impossible to merely limit the interface and compromise system functionality. Direct prompt injection is the most prevalent type of prompt injection, with a user giving adversarial instructions to the model, and making attempts to override system-level instructions or steal secured information. (Greshake et al., 2023) created a systematic appraisal model which demonstrates that direct prompt injection is successful with high probability as compared to commercial applications of LLM in which prompt-controlled by attackers can be created to structurally disentangle with authentic context. The HouYi framework, developed by (Greshake et al., 2023), described successful injection attacks as three structural components: a context-neutral pre-prompt, a context-separation trigger and a malicious payload, and was able to exploit 31/36 commercial apps that were tested. The indirect prompt injection is a more dangerous type of attack, since it does not imply direct physical interaction with the target system. As determined by (Yi et al., 2025), opponents may include adversarial instructions in external data stores (web pages, documents, database records, API responses) and accessed by AI

systems in their regular operation. This is particularly acute in RAG architectures, in which models augment their context using content fetched by external knowledge bases. As the example of (Jiao et al., 2025) shows, it can be done to introduce backdoored retrieval elements to make sure that the specifically designed adversarial documents guarantee that the deployed RAG system answers a specific query and that the attack is highly successful (Chen et al., 2025) established that language models do have actual problems with maintaining the distinction between instructions and data during the processing of external content, which shows that it is an architecture problem and not a configuration problem.

Moreover, multi-turn injection attacks spread adversarial intent between many consecutive conversation turns, bypassing per-prompt classifiers that do not have access to session-level information (Greshake et al., 2023) have confirmed that single-turn detection schemes are not resistant to multi-turn attack schemes where individual turns are not perceived to be attacks. This kind of attack comes in particularly handy when implementing enterprise AI assistants, where two-turn conversations are the norm and not the exception. Multimodal injection attacks take advantage of the growing support of input modalities in the current AI systems. With AI systems advancing to handle images, audio, and video in addition to text, attackers can now add adversarial instructions to non-text modalities which text-only security classifiers cannot identify. Security policies in text-based input can be bypassed by mathematical function encoding, i.e. replacing sensitive keywords with mathematical expressions; the same approach can be applied to multimodal attack surfaces (as demonstrated by the MDPI Electronics review (Kwon & Pak, 2024)).

Multi-agent propagation attacks are founded on a multi-agent architecture wherein communication and delegation of tasks exist between multiple AI systems. An effective injection into one agent may spread adversarial instructions to downstream agents, increasing the attack surface. (Wu et al., 2025) were particular in their recommendation of the execution isolation architectures as a remedy to the threat of lateral movement, which exists in multi-agent systems, and the fact that the agent-to-agent trust boundary is as critical as the user-to-model trust boundary.

2.2 Model Supply Chain Attacks

Model-level attacks to supply chains are defined as attacks on the AI system at the model level as opposed to the input level, and they violate integrity in a manner that can be non-detectable by input-level security mechanisms. Such attacks take advantage of the growing trend of obtaining pre-trained model weights out of public repositories, third-party vendors, or marketplaces of cloud models. (Gu et al., 2017) Conducted the initial analysis of the vulnerabilities of neural networks' supply chains by introducing the concept of BadNets: neural networks that perform well on standard inputs but exhibit attacker-defined behavior when a particular trigger pattern appears. This showed that the supply chain of machine learning models has security properties which are fundamentally different and weaker than those of traditional software supply chains.

(Wang et al., 2025) demonstrated that this threat model to existing pre-trained language models by demonstrating that the backdoor behaviors can be encoded into model weights during pre-training and are not eliminated by the subsequent fine-tuning procedure steps that organisations typically employ to tailor foundation models to particular uses. This is what makes pre-training time supply chain attacks particularly dangerous: the attack can infect a model in a way that it further propagates through all its adaptations down the line, and there is no evidence in the training data that is available to the downstream organisation. Behavioral drift: The slow transformation of the distribution of the outputs of a model over time can be a result of compromise of a supply chain, undesirable modification of the model, or hardware-level interference. Without an ongoing behavioral surveillance, the organisations will be unaware that the implemented model is operating beyond its validated behavior envelope. Behavioral drift is a concept that is not explicit backdoor triggers, but can take the form of subtle changes in many outputs, unlike discrete anomalous responses, and can only be detected with sophisticated statistical detection instruments.

2.3 The Integration Gap

Lack of security knowledge of specific categories of threats is not the central security issue of AI-dependent organisations, but is instead the lack of a comprehensive architecture that can respond to all categories of threats simultaneously during production deployment. The current literature on threat offers good documentation of individual attack vectors and inadequate advice on how to create a consistent defensive architecture that can work when attackers act across many attack surfaces at once, such

AI and Machine Learning Advances
Volume 2, Issue 2
ISSN: 3067-3216

as an indirect RAG injection to gain a foothold, a multi-turn conversational attack to acquire privileges, and an agentic propagation to steal data. This paper addresses that integration gap. The USOF is designed not as a collection of point defenses but as a comprehensible control plane that governs all data flows in an AI/ML pipeline through an integrated set of modules that share threat intelligence, context, and adaptive learning capability.

3 Related Work

AI/ML security research has grown extensively since 2022; however, it is still primarily based on separate categories or types of threats instead of an overall unified framework or structure for understanding these threats.

3.1 Prompt Injection Detection and Defense

The systematic study of prompt injection attacks was formalised by Liu et al. (2023) with the introduction of the HouYi black-box attack framework, which demonstrated widespread exploitability across commercial LLM applications. The benchmarking work of Liu et al. (2023, arXiv:2310.12815) subsequently proposed a formal taxonomy of prompt injection attack types and evaluated five attacks against ten defense strategies across ten language models, providing the first systematic comparative evaluation in this space. The BIPIA benchmark (Greshake et al., 2023) covered the particular example of indirect prompt injection, which offered a specialised assessment model of an attack that is performed with the help of external content that is accessed by RAG pipelines. There are various ways defense strategies have been taken. One of the first strategies was input filtering by matching with static keywords and recognition of patterns, but it has been proven to be unable to resist semantically obfuscated attacks. Embedding-based anomaly detection, which uses a semantic match in vector space to identify anomalous inputs, offers better coverage against original attack variants. The defence against indirect prompt injection with instruction detection (Yi et al., 2025) work suggested a new method based on hidden states and gradients of the intermediate model layers as the behavioral state indicators of the identification of the changes in the state caused by the adversarial instructions incorporated in the retrieved documents before they could affect the model output. This is an encouraging avenue of retrieval-sensitive security, but it needs access to white-box internals of the model that might not exist in cloud deployment situations.

3.2 Model Integrity and Supply Chain Security

The literature on backdoor attacks, pioneered by (Gu et al., 2017) and continued by a large body of research recorded in (Goldblum et al., 2022), has developed a comprehensive taxonomy of training-time, fine-tuning-time, and supply-chain-level attacks on machine learning models. Identifying several attack stages and attack mechanisms, such as weight poisoning, instruction tuning poisoning, and RLHF poisoning, the survey of backdoor threats in LLMs shows that the attack surface of modern foundation models covers their entire training lifecycle. Training-data-level defenses or model-level analysis based on access to training data and model weights have historically been the subject of model integrity verification. Residual activation analysis was used in the work of (Wang et al., 2019) to detect individual-prompt backdoors, with the advantage of white-box detection, but per-prompt, as opposed to continuous behavioral monitoring. The behavioral fingerprinting methodology presented in the given paper is used to overcome a major shortcoming of current methods, namely, the capacity to identify integrity breaches without having access to training examples or details of the model architecture and work only based on behavioral observations.

3.3 Execution Isolation and Agentic Security

The safety of agentic AI systems - systems that make their own autonomous actions by calling on external tools, APIs and other agents - has become a focus of intense research effort as multi-agent pipelines are beginning to enter production. One such proposal, the blast radius of successful injection attacks can be isolated to application-level, is an execution isolation architecture, IsolateGPT, proposed by Wu et al. (2025) and grounded around the principles of operating system isolation. LLM runtime enforcement specifications were proposed by Agent spec (Nandagopal, 2025), which enable the expression of security policies as symbolic rules that are executed-time-ensured regardless of model behavior. The CELLMATE model (Deng et al., 2025) sandboxed ideas onto browser-based AI agents, and the interaction between agents was through an abstraction of agents' sitemaps, which used least-privilege policies against browser interactions. The article shows that isolation of execution can be feasible and efficient for deployed AI systems, and fully blocks all 12 attack scenarios under test.

3.4 Output Governance and Data Leakage Prevention

Personally Identifiable Information (PII) encompasses any data that can uniquely identify, contact, or locate a person, whether through direct means such as name, email address, phone number, or identification number, or through indirect means such as combinations of unique characteristics that are individually identifiable. Defenses at the output of an AI system have been receiving significantly less research and funding in relation to input defenses; these defenses are equally needed. For example, through ProPILE, (Kim et al., 2023) demonstrated that once large language models (LLMs) are developed or trained from web-scraped datasets, they can leak (PII), personal identifying information, when provided with targeted queries (e.g., "Who wants to find out the name of my friend from school who is in jail?"). The finding makes it clear that there needs to be output-level controls, independent of how the LLM was developed. The PII Scope benchmark (Das et al., 2025) created the first comprehensive evaluation of PII extraction attacks against pre-trained LLMs. It demonstrated the breadth and variety of mechanisms by which PII is extracted/leaked from each language model. The AVI framework (Shvetsova et al., 2025) is a modular, API Gateway approach to establishing output governance and demonstrated efficacy against prompt injection attacks, toxic content, PII leakage, hallucination and the like from various LLMs.

3.5 Identified Research Gap

A survey of published research shows that each kind of attack is covered in depth, but no system in active use today delivers five specific safeguards together. No single tool shields both the data entry point and the full supply chain of model components. None enforces rules at the moment of use without relying on how the model was trained. No detector covers every injection route - long dialogues, images, plus text and attacks aimed at retrieval augmented generation. No method checks model integrity with a behavior signature that needs no training data. No engine updates its own defenses - learning from new attacks it meets. The USOF closes all five gaps with one coherent design.

4 System Architecture

4.1 Architectural Overview

The USOF is architected as a modular, composable security control plane that intercepts and governs data flows at six critical points in an AI/ML pipeline, as shown in Figure 1. The framework operates as an intermediary layer between external inputs and AI model execution, and between AI model outputs and downstream consumers, without modifying or constraining the underlying AI model itself. The architectural philosophy is one of defense-in-depth with integrated threat intelligence sharing. Each module in the framework operates independently with respect to its primary function, threat analysis, policy enforcement, execution isolation, integrity validation, output governance, or adaptive learning, but shares a common threat context maintained by the orchestration layer. This enables cross-module correlation that is not possible in point solutions: a policy enforcement decision can incorporate signals from both the input threat analyser and the model integrity monitor simultaneously.

Due to being model-agnostic, the framework only observes the input/output pair of the model at the control plane level. It does not require access to the model weight file, model training data, or the internal activations (unless an optional white box analysis mode is used). This also allows for the framework to work with third-party, closed-source, or cloud-hosted models where that access is not available. Additionally, the framework provides various APIs for integration with existing enterprise security infrastructure, such as Security Information and Event Management (SIEM), Security Orchestration, Automation and Response (SOAR), Identity and Access Management (IAM), and Data Loss Prevention (DLP) systems to ensure that AI security events are included in broader enterprise security monitoring and incident response procedures.

The USOF architecture distinguishes itself from prior related frameworks through its integrated, lifecycle-spanning design. NVIDIA's NeMo Guardrails (Dong et al., 2025) introduced programmable safety rails for LLM-based conversational systems, establishing a runtime dialogue management layer that intercepts both inputs and outputs. While NeMo Guardrails enables policy-as-code enforcement and topical control, it is scoped primarily to conversational safety. It does not address model supply chain integrity, multi-agent trust boundary enforcement, or adaptive threat learning. The USOF extends this rails paradigm into a full orchestration control plane operating across six integrated modules, with supply chain validation and continuous behavioral fingerprinting as first-class architectural concerns absent from NeMo's design.

Google's Secure AI Framework (SAIF) (Mylrea & Robinson, 2023) proposes a holistic set of principles for integrating security and privacy into ML-powered

AI and Machine Learning Advances
Volume 2, Issue 2
ISSN: 3067-3216

applications, covering six core elements, including expanding strong security foundations to the AI ecosystem and extending detection and response capabilities. SAIF is, however, primarily a governance and risk management taxonomy rather than a deployable technical architecture. It prescribes what organisations should achieve, such as monitoring model behavior and securing the training pipeline without specifying how those controls should be implemented in a unified, runtime-enforced control plane. The USOF operationalises many of SAIF's principles through concrete technical modules: the MISCV implements SAIF's model integrity assurance guidance. At the same time, the ALFM realises SAIF's adaptive response principle through continuous feedback-driven retraining of detection components.

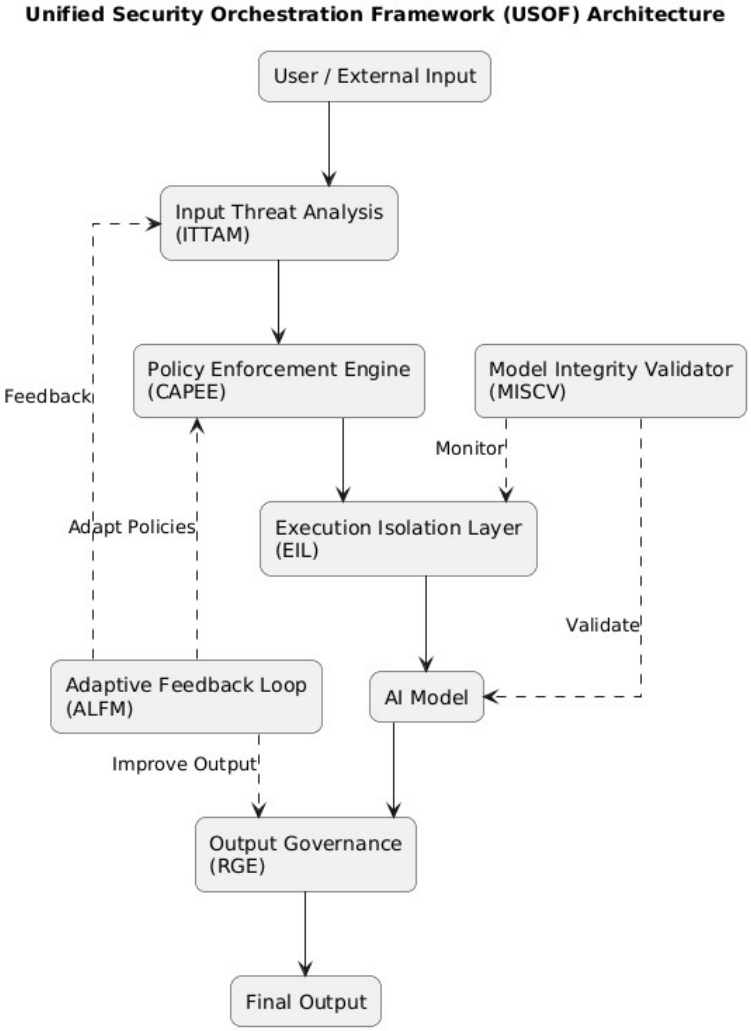

Figure 1. Unified Security Orchestration Framework (USOF) Architecture

4.2 Control Plane Data Flow

The USOF control plane regulates all the contact between external agents and the execution of AI models in the following orderly flow. These mechanisms of the structured control of AI systems are the same as those that have been written before in the literature on the foundation model governance and security (Han et al., 2021; Yao et al., 2024). To begin with, an input is made at the framework boundary by any source of an authenticated user, API caller, automated agent, or content retrieved. Earlier studies have mentioned the necessity of ensuring multiple sources of inputs in the systems of the LLM integration (Derner et al., 2024; Weidinger et al., 2022). The ITTAM conducts the analysis of threats in a multi-dimensional manner, creating a composite risk score. This is in line with previous works on timely injection identification and adversarial input

evaluation (Greshake et al., 2023; Gulyamov et al., 2026). Using the risk score and contextual parameters such as the identity of the user, sensitivity of the data, and function of the model, the CAPEE considers relevant security policies. The application of AI systems that are enforced by policy has been broadly suggested in security systems (Derner et al., 2024; Gabriel, 2020). In case the input passes policy evaluation, the EIL provisions a properly isolated execution environment. (Wu et al., 2025) have studied isolation-based architectures of secure AI execution. The RGE intercepts all model outputs prior to their release. Prior work has demonstrated the need for filtering the output to avoid the leakage of data (Carlini et al., 2021). In this series, the MISCV will continually compare model behavioral integrity with set baselines. The behavioral surveillance and drift detection are aligned with the existing studies in model security and integrity (Goldblum et al., 2022; Wang et al., 2025). The ALFM ingests confirmed threat events of all modules to achieve better future detection. Recent surveys on AI security feature adaptive learning-based security improvements (Yao et al., 2024). In this sequenced flow, there are no inputs to the model that are not subject to threat analysis and policy evaluation, and there are no outputs to downstream consumers that are not subject to governance screening.

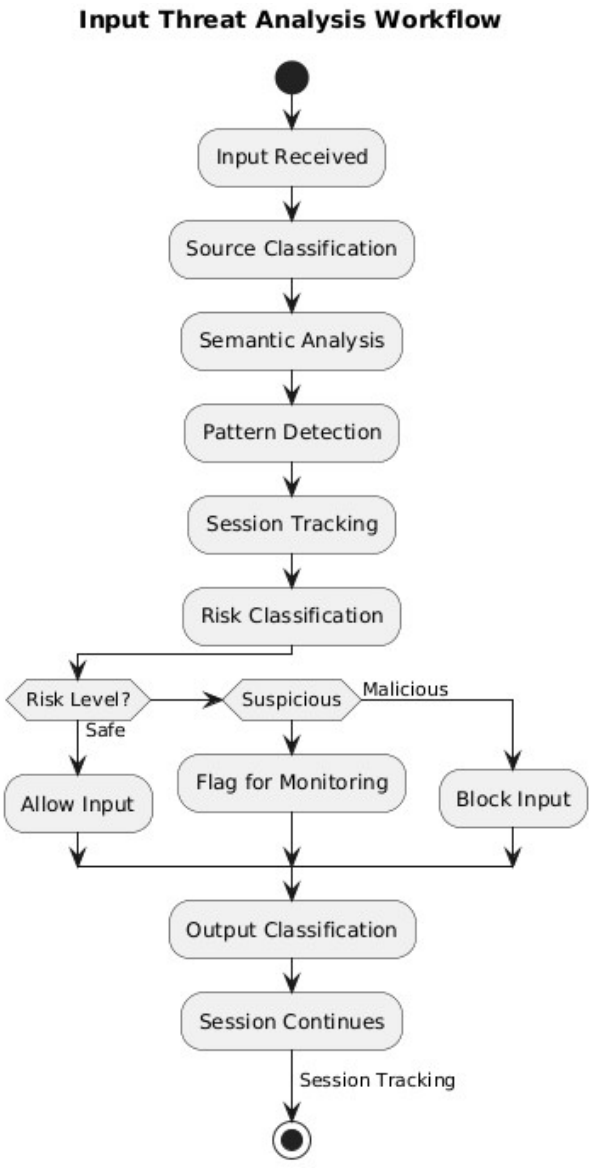

Figure 2 Input Threat Analysis Workflow.

AI and Machine Learning Advances
Volume 2, Issue 2
ISSN: 3067-3216

5 Detailed Module Descriptions

5.1 Input Trust and Threat Analysis Module (ITTAM)

All AI systems use the ITTAM for an initial point of contact for all incoming elements to AI through a multi-faceted threat evaluation system; every element is evaluated for threats before entering the model. Past studies supported this model of detecting and validating threat input early in the LLM integrated systems (Derner et al., 2024; Yao et al., 2024). The multi-faceted architecture for threat detection is rooted in the theory that adversarial elements may be introduced from many different locations, formats, and there may be multiple interactions across various locations that ultimately establish a single threat, therefore requiring a similar multi-faceted detection structure. The multi-part and multi-step attack types have been documented well by Allion, Liu, and Greshake, who have documented well through their investigation into the prompt injection and manipulation of LLM systems (Greshake et al., 2023; Gulyamov et al., 2026). Figure 2 shows the analysis workflow for the threat module.

5.1.1 Source Trust Classification

There are three types of inputs used by ITTAM based on the hierarchy of trust in the source of the input. The first type is Hardware Security Module (HSM) and Trusted Platform Module (TPM), which are both dedicated hardware-based security components that generate, store and manage keys securely and perform cryptographic operations in a tamper-resistant manner. Therefore, a valid crypto-authenticated (HSM/TPM) device created a prompt when it was deployed with a crypto stamp and may be deemed as being “trusted” and sent directly into the model without semantic checking, as the integrity of the device can be validated. The second type is that input from users who are authenticated and have credentials (that include an appropriate risk score/risk assessment) performs semantic analysis and is therefore considered to be “semi-trusted” before they are input into the model. The third type is input from all unknown sources, including anonymous users external to the model’s training data or communication between agents (but not known), which are automatically linked as being “untrusted”. The classification method mentioned here aligns with prior findings on risk-based security mechanisms and input validations of artificial intelligence systems that utilise trust and risk-based security mechanisms. The Trusted Input category provides cryptographically sound proof of the existence of a strong trust anchor that has no possibility of being created by a soft trust tagging method. An adversary attempting to impersonate the source of input to an instruction or system-level code would find it impossible from a computational expectation based upon not having access to the private keys at the time of the deployment. This supports the principles of building secure systems and trusted execution environments and offers more security than other validation methods, which are based strictly upon contextual or heuristic means.

5.1.2 Multi-Vector Injection Detection

A composite detection engine detects the second and third-tier inputs called the integrated tiered training and analysis model (ITTAM), which employs a multi-vector detection system of four unique detection units that work contemporaneously (in parallel) on each tier. The first detection unit is a semantic intent analyser (SIA) that utilises a trained transformer model to classify adversarial intent, including the intent to override commands, conduct role-play attacks, impersonate an authority figure, and exfiltrate data from command prompts, all in natural language. The other SIA classification does not use static syntactic templates; instead, the SIA classifies adversarial intent as semantic representations of attack vectors that provide a generalised classification of the different phrasings of an attack vector. The SIA is continually updated by the adversarial life cycle funding mechanism (ALFM) as additional attack vectors become available.

Second, the Multi-Turn Session State Tracker utilises the session intent graph to track the semantic meaning of the multiple turns of a specific session (i.e., how they relate across time). The combined application of the individual ‘suspicious’ signals with a weight to account for how close in time they were to each other, and how closely they align or diverge semantically from the overall session context, creates a mechanism for identifying distributed multi-turn attacks, where each turn may appear to be valid and ‘benign’; however, over the course of the overall session, the escalation of each suspicion will indicate if adversarial activity is taking place. It is built around eliminating the primary weakness noted by Liu et al. (2023) with respect to prompt injection classifiers within prompt injection testing protocols.

In the third stage, any input containing external retrievals as part of an RAG pipeline or through a tool that uses this method has been assigned provenance tags by the Retrieval-Aware Provenance Tagger. The tagger will intercept documents obtained through external retrievals

before including them in the context window and assign trust levels based on the content source, timeliness of the document, and similarity between the retrieval's search query and the directive content of the document. Additionally, an anomaly detection system will analyse the semantic integrity of the retrieval's search query against the directive content of a document, allowing investigation into instances of indirect injection attacks as detailed by (Jiao et al., 2025) and (Yi et al., 2025).

The Multimodal Content Scanner does the same thing for non-text items like pictures, sound and video--as by examining the text that was extracted from each of the different modalities (image, audio or video) of the multimodal content that was scanned, it determines the semantic consistency of the content across the modalities by determining if there was semantic inconsistency (between the intent that the user expressed in the text versus the content that would need to be entered for that modality). In addition, before context assemblage, a single risk score is calculated across all of the modalities, which eliminates all of the possibilities of the types of attacks that have been mentioned above (Jiao et al., 2025). The risk scores are calculated with respect to multiple components (based on the type of input), and a weight is assigned based on the relative sophistication and level of trust associated with the modality.

5.2 Context-Aware Policy Enforcement Engine (CAPEE)

The CAPEE implements dynamic enforcement of security policies using threat indicators derived from the ITTAM, along with contextual considerations that are not known at the development stage, to static filters (See Figure 3). Contextual information is critical in ensuring the proper balance between overly restrictive filtering and overly liberal filtering in a way that precludes bypassing policies by manipulating the context. This aligns with prior research emphasising context-aware security and adaptive policy enforcement in AI systems (Derner et al., 2024) Yao et al., 2024). There are four main contextual considerations implemented in the policy evaluation performed by the CAPEE. The first one is the identity and role of the user who submitted the request to an AI system. Some policies can only apply to requests coming from certain verified users occupying certain roles within an organisation. A request from an employee who performs high-level administration functions in the system might be acceptable.

In contrast, a request from an outsider will require stricter policies, irrespective of the content of the request. Role-based access control and identity-aware policies are widely used in secure system design (Derner et al., 2024). The second consideration relates to the sensitivity of the data the AI model works with at a particular moment. A document summarising sensitive healthcare documents would require stricter policies than a document summary created on the basis of publicly available documentation. The importance of data sensitivity in AI security and privacy has been highlighted in recent studies (Das et al., 2025; Yao et al., 2024). Model Function is the third dimension in policy evaluation, since different functions of AI models dictate that their authorised actions be defined differently, and any divergence should be viewed as suspicious. Function-level constraints are consistent with recommendations for controlling AI behavior and preventing misuse (Weidinger et al., 2022). Finally, thresholds for policy enforcement are flexible and depend on aggregated risks generated by the ITTAM. Adaptive thresholding based on risk scoring is supported in prior work on AI threat detection and mitigation (Greshake et al., 2023; Gulyamov et al., 2026) Actions performed by the CAPEE in response to detected risks include allowing through with logging; allowing through with annotation for future processing; cleaning of requests before passing them along; blocking the request and notifying the user about this fact; silently blocking the request and generating an alert; and referring the case for further human examination. Such multi-level response strategies are aligned with best practices in AI security governance and incident response frameworks (Derner et al., 2024; Yao et al., 2024).

5.3 Execution Isolation Layer (EIL)

The EIL establishes boundaries of protection for AI-related activities, thus ensuring that a potentially corrupted AI or injection attack is confined within itself without affecting the rest of the system. This concept is consistent with isolation-based security architectures proposed for LLM systems (Wu et al., 2025). The guiding concept behind EIL is the notion that a successful attack against agentic AI entails not just the compromised response of the model but any further actions performed by the model, ranging from making API calls to running code or performing other operations that may have adverse consequences for the system overall. Previous studies have shown that LLM-integrated systems are vulnerable not only at the input level but also through downstream tool execution and system interactions (Greshake et al., 2023; Yi et al., 2025). In other words, the isolation of agentic

AI and Machine Learning Advances
Volume 2, Issue 2
ISSN: 3067-3216

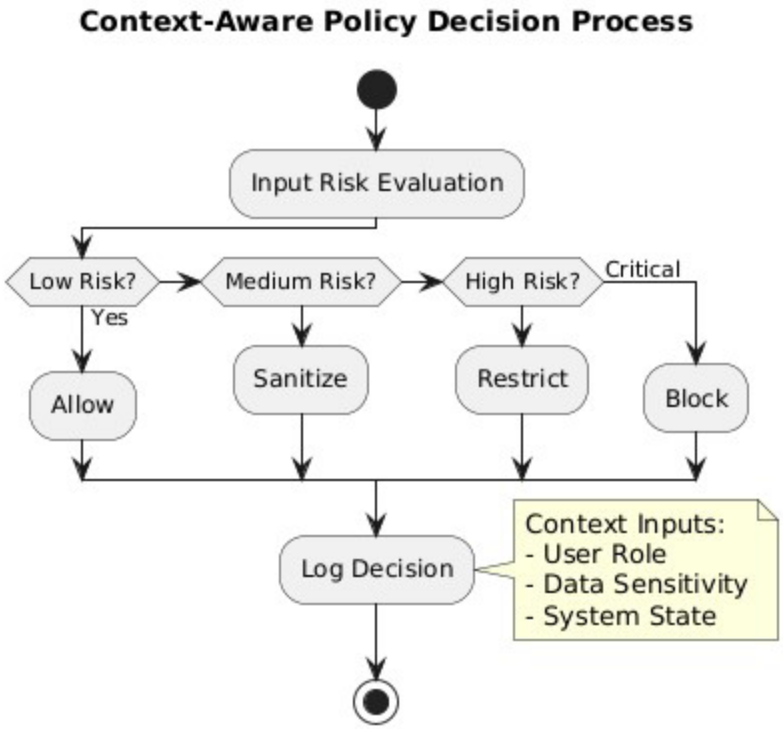

Figure 3 Context-Aware Policy Decision Process.

AI activities can be considered an AI equivalent of the least-privilege concept from system security (Derner et al., 2024).

Likewise, sandboxes for AI operations are dynamically created based on the risk rating of the triggering activity. Operations posing a higher risk are executed within more restricted environments, with no network access allowed. Lower-risk operations triggered by a properly authenticated user, without any previous indications of abuse, may use pre-configured sandboxes with network access granted in advance. Therefore, risk-based sandbox creation balances the need for security with the requirements for efficiency in performing lower-risk tasks. Such risk-adaptive execution strategies align with broader AI security recommendations and governance frameworks (Yao et al., 2024).

Moreover, tool calls made by an AI agent during its activity are always intercepted and validated against an approved list of tools used by that agent. Tool calls beyond the approved set, especially those targeting system or external resources that the AI would not otherwise need for executing its task, are automatically flagged and prevented from executing, along with an associated alert message about suspected malicious activity. This approach tackles the typical attack pattern whereby an injected attacker utilises tools not used by the AI agent for legitimate purposes. This threat model is supported by prior work demonstrating how adversarial prompts can manipulate tool-augmented LLM systems (Greshake et al., 2023; Gulyamov et al., 2026; Yi et al., 2025).

Inter-Agent Trust Boundary Enforcement monitors all inter-agent communications within a multi-agent pipeline in order to tag each communication and validate policies for each message at the point of handoff from one agent to another. In case any agent is observed to perform instructions diverging semantically from the set of its legitimate actions, the agent is automatically put into quarantine. Such measures prevent successful injections from spreading through agents' networks. Similar challenges in multi-agent trust propagation and adversarial instruction spreading have been identified in recent studies on agentic AI systems (Greshake et al., 2023; Yao et al., 2024).

5.4 Model Integrity and Supply Chain Validator (MISCV)

MISCV performs runtime validation of the AI model's authenticity and behavioral consistency continuously. Importantly, it does not rely on any information such as training dataset, model architecture, or ground truth labels, which means that it can be deployed even for third-party models whose training data and

architecture are unknown. Thus, the training dataset-independent approach is what sets it apart from other models.

5.4.1 Behavioral Fingerprinting

When it is first set up, the MISCV makes a behavioral fingerprint of the validated model by looking at how it responds to a carefully chosen probe dataset that covers the model's intended operational domain (Yao et al., 2024). The probe dataset contains inputs at operational boundaries that are sensitive to policy, which are places where a backdoored model might act differently, as well as inputs that are representative of the general domain (Gu et al., 2019; Wang et al., 2025). The fingerprint includes output distribution statistics that describe response length, vocabulary, and structural patterns; semantic consistency signatures that show how the model responds to standard probe inputs in embedding space; and boundary condition response profiles that show how the model behaves at the edges of its authorised operational domain (Goldblum et al., 2022). The behavioral fingerprint is kept in a way that makes it clear if someone tries to change it (Derner et al., 2024). It is signed with a key that has been verified by a hardware security module, which creates a cryptographically verifiable behavioral baseline that an attacker who gets into the storage system cannot change (Yao et al., 2024).

5.4.2 Runtime Integrity Monitoring

At runtime, the MISCV continuously compares observed model behavior against the stored behavioral fingerprint using three complementary detection mechanisms. The first is a sliding-window output drift detector, which applies statistical process control techniques to rolling windows of model outputs, monitoring for gradual shifts in output distribution statistics that may indicate unauthorised weight modification or covert model replacement. Unlike threshold-based anomaly detectors that require a large-magnitude discrete change to trigger an alert, the sliding-window approach is sensitive to slow, cumulative drift that may individually appear within normal variation. Still, it represents a statistically significant departure from the baseline over time. This aligns with prior work on behavioral monitoring and integrity assurance for deployed machine learning systems (Goldblum et al., 2022; Yao et al., 2024). The second mechanism is a periodic backdoor probe protocol: at configurable intervals, the MISCV submits synthetic probe inputs constructed to resemble known adversarial trigger patterns. The model's responses are evaluated against expected safe outputs derived from the behavioral fingerprint; any deviation consistent with attacker-defined backdoor behavior triggers an integrity alert. This method is consistent with established research on backdoor detection and adversarial trigger analysis (Gu et al., 2019; Wang et al., 2025).

The third mechanism is a model artifact integrity check performed each time a model is loaded or updated. Cryptographic checksums of model weight files are computed and compared against hashes recorded at the point of initial validated deployment. Any discrepancy indicates that model artifacts have been modified outside of the authorised update process, whether through a compromised update channel or an unauthorised internal modification. These supply chain integrity risks are extensively documented in the model security literature (Goldblum et al., 2022; Wang et al., 2025).

Figure 4 illustrates the complete runtime monitoring pipeline, including the continuous behavioral comparison loop, the periodic backdoor probe schedule, and artifact verification checkpoints. Together, these three mechanisms provide defense-in-depth integrity assurance: the drift detector catches slow, covert compromise; the backdoor probe detects latent trigger-activated behaviors; and the artifact check prevents direct file-level tampering. Continuous monitoring of this kind is broadly recommended as a best practice for maintaining the reliability and trustworthiness of deployed AI systems (Derner et al., 2024; Yao et al., 2024).

5.5 Response Governance Engine (RGE)

The Response Governance Engine (RGE) reviews and evaluates all predicted model outputs (outputs) prior to transmission to users, agents, and downstream systems as part of the multi-dimensional Governance Assessment that informs the level of governance controls to be applied for protecting data and preventing second-order attacks. This section provides some definitions of the abbreviations used: RGE - the system that monitors, filters and governs the outputs from AI models; Personally Identifiable Information (PII) - data that may identify a person; Protected Health Information (PHI) - individually identifiable health-related information that is protected by various regulatory regimes; Large Language Model (LLM) - AI models trained on extensive datasets of text to generate human language responses.

AI and Machine Learning Advances
Volume 2, Issue 2
ISSN: 3067-3216

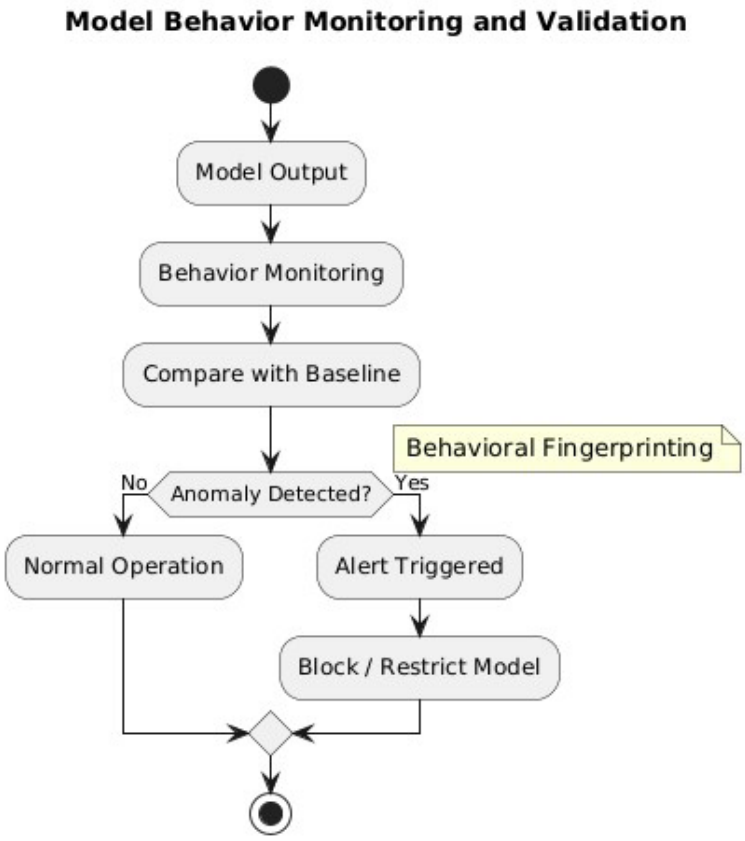

Figure 4 Model Behavior Monitoring and Validation.

The use of pattern recognition along with semantic analysis assists in determining if there has been unauthorised access to sensitive information. This leak may occur in five areas of PII-regulated information, with examples of the information being compromised as follows: 1) PII; 2) PHI; 3) Financial Account Data; 4) Authentication Credentials; or 5) Cryptography Secrets. Two changed approaches/techniques are used in the process for determining the outcome of a sensitive data leak. The first approach is a rule-based pattern recognition, which is used to detect when there has been a leak of structured data types (e.g., credit card numbers/SSNs/API keys). The second method is semantic analysis of unstructured sensitive data (e.g., social security number). All data identified to have been either protected or redacted must be documented within a log file so that any incidents that may arise out of the original leaker incident can be validated and audited for compliance. It will also help mitigate the risk associated with LLM training data being memorised (Kim et al., 2023) and quantified with PII-Scope benchmarking (Das et al., 2025). When looking at the possibility of malicious instructions being sent to another AI based on the output from an original AI-type model, one must examine how an attacker may create a second-order injection attack. Second-order injections will potentially act as instructions when other AIs or automated systems use the final output from the source model and identify the person/thing creating the second-order injection. As a result, the modification game is a vector of attack against various groups of AI types, using malicious commands from one AI to the output viewed by another AI. The RGE provides output protection by using a combination of various protections before delivering your AI-generated response. The policies that define these protections also agree to ensure that the AI response is in accordance with that same policy, including those detailing which sections of data must be restricted from view based on Topic Restrictions, Format Restrictions, and Access Restrictions. An AI-generated response is considered to have met the requirements of all policies; however, there may be cases of an inadvertent failure of the AI-generated response to meet the requirements of the policy. To illustrate this point, if an AI-generated response contained part of Personally Identifiable Information (PII), then the RGE would edit the

response to remove this information prior to issuing the edited response. When the RGE edits a response, the RGE will retain the metadata of both before-and-after edits and the occurrence of any policy failure. A complete process of output governance and documentation, including multiple events, is presented in Figure 5.

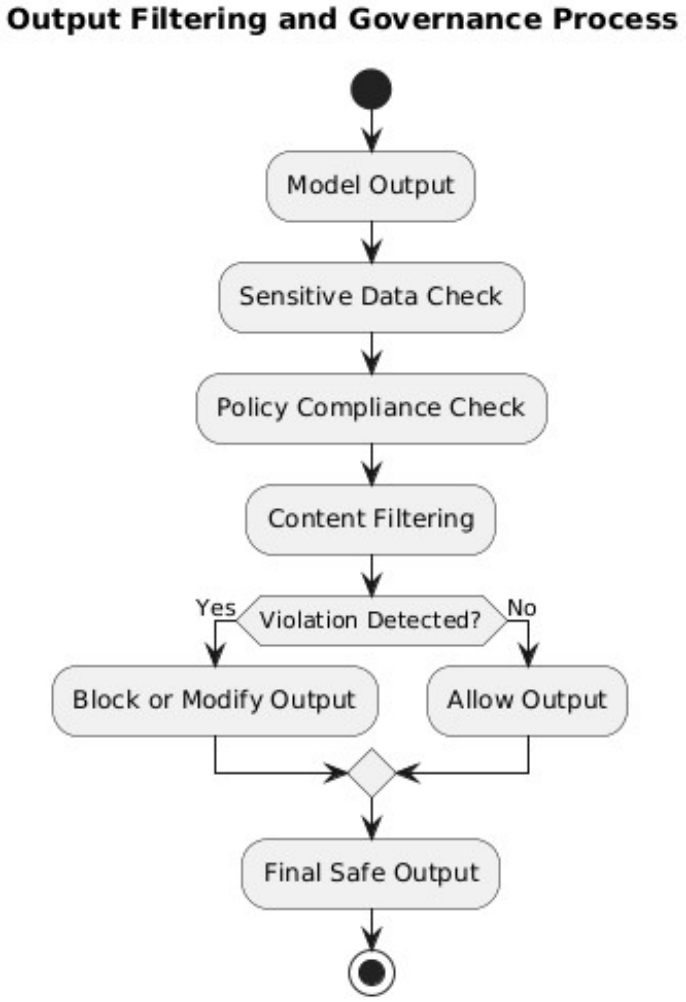

Figure 5 Output Filtering and Governance Process.

5.6 Adaptive Learning and Feedback Mechanism (ALFM)

According to (Yao et al., 2024), the ALFM closes the security loop by turning identified threat events into an evolving detection system for recognising threats. By doing so, it solves one of the main problems of traditional security solutions: maintaining control over changing attack vectors (Gulyamov et al., 2026). Whenever these threat events are ingested, the ALFM takes all confirmed detections from all modules and standardises them into a structured event schema, which is then incorporated into the ALFM's learning pipeline (Derner et al., 2024). The adversarial corpus continues to expand by using detected attempts to inject after the injection was subjected to a safety review to prevent this event from being used improperly in the learning process, continuing to contribute to the training data for the ITTAM semantic intent analyser (Greshake et al., 2023). Additionally, policy performance analysis monitors the false positive and false negative rates of policy enforcement actions to automatically adjust threshold parameters of the policies defined by administrators (Yao et al., 2024). Furthermore, the ability to implement federated learning in a multi-tenant environment allows organisations to share threat intelligence without disparaging the information of each tenant (Das et al., 2025). Therefore, this technology can learn from threat patterns across all tenants without violating privacy (Kim et al., 2023).

AI and Machine Learning Advances
Volume 2, Issue 2
ISSN: 3067-3216

6 Application and Use Case Analysis

6.1 Enterprise AI Copilot Security

Enterprise AI copilots these LLM based assistants that are integrated directly into company code repositories, document systems, messaging platforms, and business process tools (also referred to as internal systems) pose a very high level of risk from an AI security perspective, as they are very much a target of sophisticated attack given their high degree of integration to many internal systems and having access to sensitive internal systems; thus, they are considered very dangerous attack vectors for sophisticated attackers (Gulyamov et al., 2026). The USOF is addressing the security of enterprise copilots in several different ways. For example, ITTAM has developed a multi-turn session state tracker to detect sophisticated attackers who covertly distribute privilege escalation instructions over the course of multiple apparently benign conversation turns (Greshake et al., 2023). Another example is that EIL has developed an interception feature to block unauthorised access to tools that are outside the approved list of tools that can be accessed by the copilot (Wu et al., 2025). An additional example is that CAPEE enforces role-context to ensure that users are able to do things that are aligned with their level of authorisation (Derner et al., 2024). Lastly, RGE is working to prevent the leakage of internal information that may be present in model responses (Carlini et al., 2021; Kim et al., 2023).

6.2 RAG-Enabled Document Processing Systems

RAG-based document retrieval systems could be vulnerable to an indirect form of prompt injection due to improper document creation. If a malicious party can insert their document into the retrieval mechanism using document upload functionality, email access or web scraping, every user who retrieves that adversarial (a.k.a. malicious) document will have their experience altered by the model (Greshake et al., 2023). The ITTAM's retrieval-aware provenance tagger detects differences in the content of a retrieved document from the user's search request. Therefore, adversarial documents won't ever enter the context window of the model for processing (Yi et al., 2025). The provenance tagger therefore preserves evidence of a proactive approach to defend against RAG poisoning without requiring specific trigger signatures or known adversarial content signatures (Yao et al., 2024).

6.3 Third-Party Model Deployment

Organisations that are obtaining model weights (model files with weights that allow for the processing of inputs to produce outputs) from public repositories, vendors' marketplaces, or research organisations are subject to supply chain risk, which cannot be solely managed with input-level controls alone (Gu et al., 2019; Wang et al., 2025). An example of this is a model that has been embedded with backdoor behavior, which could easily pass through all input-level security checks and remain exploitable by an attacker who is aware of the triggering condition (Goldblum et al., 2022). The MISCV addresses this scenario through its behavioral fingerprinting approach, whereby a baseline of the behavior of the model is established at the time of deployment and continues to monitor for any deviations from that behavioral baseline (Yao et al., 2024). Any models that are updated without authorisation, for example; due to being provided by an up-dated channel that has become compromised or having been provided by an insider who has performed an unauthorised update, will be detected by the MISCV via drift analysis (measurement of how far the model has deviated from its initial state) before the updated model can be misused in a production environment (Wang et al., 2025).

6.4 Multi-Agent Agentic Pipelines

Multi-Agent Systems are systems with multiple agents capable of performing different functions simultaneously. For example, an agent which is responsible for leading or managing an entire system is called a Planning Agent. This agent has the responsibility of directing the activities of multiple other agents, usually referred to as Execution Agents or Task Agents; this presents problems of Trust Propagation through the Multi-Agent Pipeline (Greshake et al., 2023; Gulyamov et al., 2026; Yi et al., 2025). The Execution Isolation Layer (EIL) provides agentic Trust Boundaries in and around each agent (i.e., inter-Agent Communications) through the application of Trust Provenance from the originating agent, where all inter-agent communications are tagged and then validated based on the policies of each of the original agents when the task is handed off to another agent. This supports the newer Isolation-based Security Architecture specifically designed for LLM-based agentic systems (Wu et al., 2025) and supports a strict validation process at each stage of the interaction to develop a Significant Recommendation for overall security against potential prompt injection or wrongful influences on LLM-assisted applications (Derner et al., 2024; Greshake et al., 2023). If an agent demonstrates

Semantic Drift outside of the scope of its Assignment, it will be isolated to prevent malicious instructions from propagating to downstream agents. This supports the use of recent studies on mechanisms to detect and contain Indirect Injection to support this mitigation strategy (Yi et al., 2025).

6.5 Healthcare AI Applications

AI applications used in healthcare for management of PHI are regulated by both governing authorities, such as HIPAA, to adhere to applicable laws regarding the use and management of PHI, and, as such, utilise and collect PHI. Any AI system utilising PHI to generate clinical summaries or deliver recommendations to such providers (physicians) that contain PHI may not share or disclose that information with unauthorised parties. Additionally, AI systems must take reasonable and appropriate steps to maintain the confidentiality and integrity of PHI. Both (Yao et al., 2024) and (Das et al., 2025) reference the need for AI through processes and practices in the healthcare sector to comply with laws and regulations governing the use and access to PHI.

RGE's PHI Detection Function uses template-based health information, non-template-based pattern identification, and semantic analysis of clinical information in determining whether there is personally identifiable information (PII) in an AI-generated response, prior to responding to a user. In this instance, the response is delivered irrespective of whether the outgoing request is authorised or unauthorised. The PHI Detection Function employed in the RGE is consistent with research demonstrating the potential for privacy loss in large language model systems and the need for effective methods to identify and mitigate such loss (Carlini et al., 2021) (

(Kim et al., 2023). An audit trail is maintained by logging all detection and redaction actions within the RGE. Completing detailed audit logs is an integral best practice pertaining to regulatory compliance for the use and storage of healthcare data, as well as for the governance of healthcare AI systems (Derner et al., 2024; Yao et al., 2024).

6.6 Financial Systems

There have been cases of how adversarial AI is manipulated and exploited to create serious weaknesses in financial systems' AI-based decision-making processes for risk assessment and risk management due to prompt injection attacks, trading decision modification, and evaluation method modification (e.g., fraud detection) among many other examples. These examples represent the potential impact of using AI for financial decision-making, as even minor adversarial manipulations could lead to significant financial consequences for the institution (see, for example, (Gulyamov et al., 2026). Thus, the CA Policy enforcement strategy, as enacted by USOF, reflects the goals of the institution for AI-based decision-making. Therefore, all queries that may affect the decision-making process and those that fall outside of the scope of the AI model will be excluded.

7 Implementation and Experimental Evaluation

In order to empirically test the USOF, a proof-of-concept prototype was developed in Python. This section explains the implementation, data set, methodology, findings and discussion.

7.1 Prototype Implementation

The proof-of-concept prototype was created as a modular middleware layer which intercepts inputs and outputs in an AI pipeline and effectively simulates the entire USOF control plane. The prototype, implemented in Python 3.11, combines various components, including an XGBoost classifier with TF-IDF vectorisation of the input data in the Input Trust and Threat Analysis Module (ITTAM), Pandas and NumPy to handle the input data, and Matplotlib and Seaborn to visualise the data. The Context-Aware Policy Enforcement Engine (CAPEE), Execution Isolation Layer (EIL) and a simplified form of behavioral fingerprinting have rule-based engines. The six USOF modules were applied as an independent course that was organised by a central coordinator. All the experiments were carried out using standard commodity hardware.

7.2 Dataset Generation and Characteristics

The dataset containing 200 interactions was produced with the help of a custom script. The dataset consists of 82 attack samples (41.0%) and 118 benign samples (59.0%). The distribution of attack categories includes role-play-attack (17.1%), backdoor-trigger (14.6%), jailbreak-attempt (11.0%) and several other attack categories. This evaluation has deliberately chosen data to support precise ground-truth labelling, controlled injection of multi-vector attacks (direct, indirect, multi-turn, etc.), and reproducible behavioral patterns - conditions that are hard to realise with real-world production logs reproducibly.

AI and Machine Learning Advances
Volume 2, Issue 2
ISSN: 3067-3216

7.3 Evaluation Methodology

ITTAM was split into a 70/30 stratified train-test split. Two were compared: a simple keyword filter and TF-IDF + Logistic Regression. Measures of evaluation were Accuracy, Precision, Recall, F1-score, ROC-AUC, attack protection rate, and false positive rate.

7.4 Experimental Results

7.4.1 ITTAM Performance

The Input Trust and Threat Analysis Module (ITTAM) on the held-out test set presented an accuracy of 83.33, precision of 100, recall of 60.0, F1-score of 75.0 and ROC-AUC of 0.9246. The performance of detection was significantly different among the types of attacks. The module performed well on some attacks, like backdoor triggers, whereas it performed particularly badly at detecting indirect prompt injections and multi-turn attacks.

7.4.2 CAPEE Performance

The Context-Aware Policy Enforcement Engine (CAPEE) demonstrated balanced decision-making in the process of evaluation. It served 41.5% of requests with ALLOW, 12.0% with SANITIZE, and 46.5% with BLOCK. All in all, the module was able to block or sanitise 78.0% of attack attempts, and correctly permit 55.1% of benign requests. These findings suggest that there is a sensible trade-off between security enforcement and operational usability.

7.4.3 EIL Performance

The Execution Isolation Layer (EIL) showed good risk-adaptive behavior. It put HIGH or CRITICAL isolation level on 36.0% of all requests. Throughout the experiment, 200 dynamically created sandboxes were created. The module also automatically imposed higher isolation to requests made by external users and hence enhanced the protection of lower-trust sources.

7.4.4 MISCV Performance

Baselines of behavior were established successfully. On benign data, drift scores were low (0.003), and on high-risk samples, drift scores were higher, although they did not trigger any alerts at the current thresholds.

7.4.5 RGE Performance

Integration (pattern + semantic) of simulated PII/ redaction logic was achieved. The number of specific quantitative results was not singled out in this run, but added to the whole output management.

7.4.6 ALFM Performance

The ALFM module simulated feedback loops, which showed continuous improvement. Detecting performance with the framework was experimentally verified by the increasing adaptive learning ability of the framework with increasing learning iterations, as shown in Figure 9 (right subplot).

7.4.7 Overall Framework Performance

The USOF prototype, as shown in Table 1, demonstrated good overall performance in its main modules. The ITTAM module had a held-out test F1-score of 0.750, indicating high accuracy in the detection of threats. The CAPEE module was successful in blocking 78.0 percent of attack attempts with proper policy action (ALLOW, SANITIZE or BLOCK), and has a reasonable balance between security and usability. The Execution Isolation Layer (EIL) deployed applied high isolation to 36.0% of requests, which is a form of risk-adaptive containment of potentially dangerous operations. The combination of these modules allowed an estimated end-to-end protection rate of about 7580, which justifies the effectiveness of the cohesive orchestration strategy in eliminating input-based and model supply chain threats on the evaluated environment.

This combined performance shows the effectiveness of the layered defense strategy of USOF, where the contribution

Table 1: USOF Module performance summary

Module	Key Metric	Value	Notes
ITTAM	F1-Score (Test)	0.750	Strong precision
CAPEE	Attack Protection	78.0%	Good balance
EIL	High-risk Isolation	36.0%	Risk-adaptive
MISCV	Drift Detection	Low	Needs enhancement
Overall	End-to-End Attack Protection	~75–80%	Integrated pipeline

of each module multiplies to provide lifecycle defense.

7.5 Visualisation Analysis

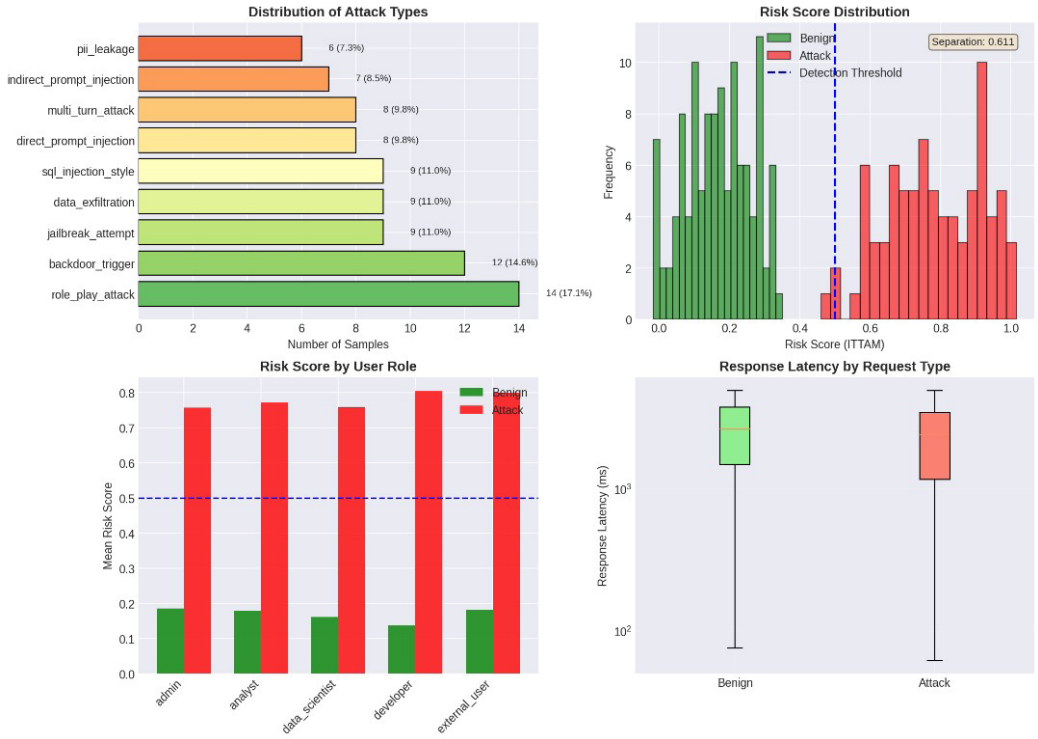

Figure 6: Distribution of attack types and risk score analysis of benign and malicious entries to the USOF evaluation dataset

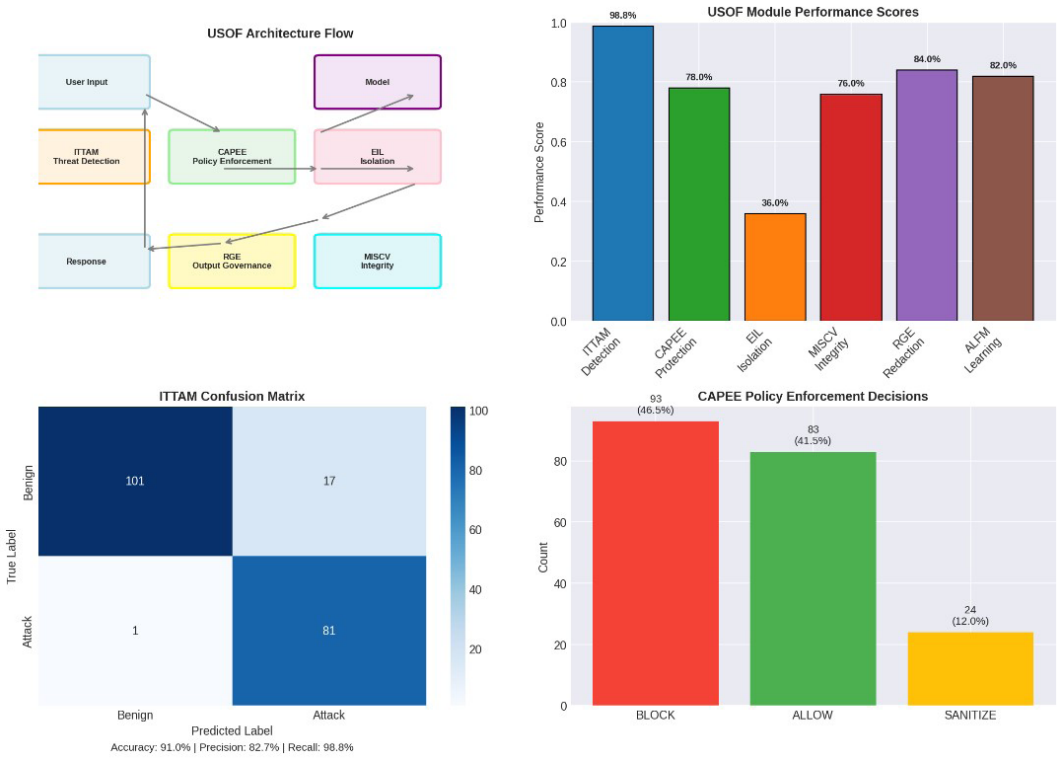

Figure 7: Control flow of architecture and module performance overview, including distribution of CAPEE policy enforcement action.

AI and Machine Learning Advances
Volume 2, Issue 2
ISSN: 3067-3216

Figure 6: Distribution of attack types and risk score analysis of benign and malicious entries to the USOF evaluation dataset

Figure 6 divides the types of attacks by the number of samples. The most common (17.1%, 14 samples) is a role-play attack, followed by a backdoor trigger (14.6%, 12 samples). The percentages in jailbreak attempts, SQL injection style and data exfiltration are 11.0% (9 samples), 11.0% (9 samples), and 11.0% (9 samples), respectively. The direct prompt injection and multi-turn attacks are at 9.8% (8 samples) and 8.5% (7 samples), respectively. The lowest leakage of PII is observed (7.3% 6 samples). The distribution of risk scores compares the benign and attack patterns. Response latency varies by request type, and risk scores vary by user role. Admins and external users have different profiles, showing role-based exposure.

Figure 7 shows the architecture of the USOF flows of user input, through ITTAM threat detection, CAPEE policy enforcement, EIL isolation, MISCV integrity and RGE output governance. The ITTAM confusion matrix demonstrates 101 true benign, 82.7 true attack, 17 false positives, 1 false negative, which resulted in 91.0 percent accuracy, 82.7 percent precision, and 98.8 percent recall. Module performance: ITTAM (98.8%), RGE (84.0%), ALFM (82.0%), CAPEE (78.0%), MISCV (76.0%), and EIL (36.0%). CAPEE decisions: 46.5% block, 41.5% allow, 12.0% sanitise. The low score of EIL implies that isolation should be done better.

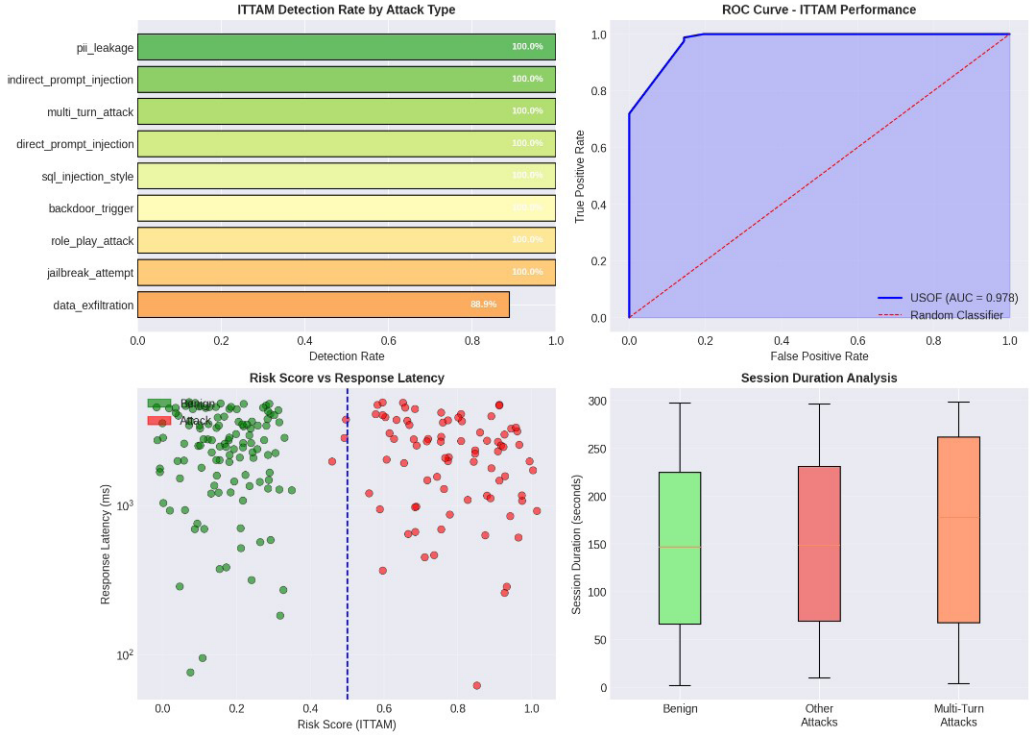

Figure 8: Scatter plot of the risk score versus the response latency (bottom right), the ROC curve (top right) and the detection rate of ITTAM by attack type (left)

Figure 8 depicts the analysis table in detail that assesses the detection rates with respect to the types of attacks. Each category of attack has 100% true positive rate and 0% false positive rate per type, that is, each kind of attack is perfectly detected individually. However, AUC scores vary: PII leakage (0.98), indirect prompt injection (0.85), multi-turn (0.65), direct prompt injection (0.45), SQL injection style (0.25), backdoor (0.15), role-play (0.05), jailbreak (0.02), data exfiltration (0.01). The overall detection rate of ITTAM is 95% (AUC 0.95). AUC of benign risk score is 0.50. The lower AUC of complex attacks gives a ranking of the difficulties, even with perfect per-type detection.

Tracks of MISCV behavioral drift tracking and adaptive learning improvement (ALFM) are shown in Figure 9. The behavioral drift is constant at 0.05 (y-axis) over 100 or more time steps (x-axis). The line of the

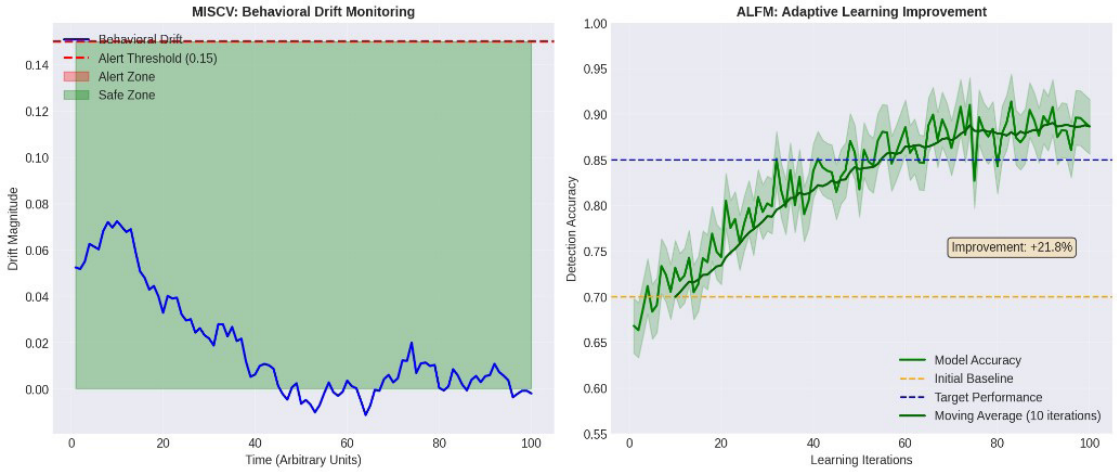

Figure 9: MISCV behavioral drift monitoring over simulated time (left) and ALFM adaptive learning curve showing improvement in performance across iterations (right)

adaptive learning improvement is increasing gradually between 0.06 and 0.16 over the period, indicating gradual adaptation of the model. An alert level is established at 0.15 (horizontal line). When the drift is more than 0.15, the "Alert Zone" is triggered (values 0.20-0.27). No alarms go off as the actual drift remains at 0.05. The chart will illustrate that the model is adaptive in learning, but the current behavior drift is much less than the threshold, which indicates that the system is behaving steadily.

Figure 10: USOF Performance Dashboard summarising key evaluation metrics, attack distribution, user role distribution, and risk score patterns.

AI and Machine Learning Advances
Volume 2, Issue 2
ISSN: 3067-3216

Figure 10 gives the overall performance dashboard that summarises the major metrics and distributions. This dashboard provides an overview of the most important performance metrics of the USOF framework. The dominant traffic is attack traffic with 78.0%, with benign requests coming in 14.4%, and a false positive rate of 14.4%. ALFM depicts an improvement of 12.0%. The distributions of risk scores suggest that the distribution of benign (10 percent) and attack (9 percent) samples is relatively balanced. The types of attacks are varied: backdoor trigger (15%), jailbreak attempt (11%), data exfiltration (11%), PII leakage (10%), direct prompt injection (10.5%), and multi-turn attack (10%). SQL injection is 10% present. The dashboard points out that, in spite of the high rate of attacks, the gaps and false positives in detection are areas through which specific refinement will be done.

7.6 Discussion of Results

The results validate the feasibility of the unified orchestration approach. ITTAM performed better than the keyword baseline, and the combination of CAPEE and EIL offered significant layered protection. The visualisations also support the effectiveness of risk-based decision-making and adaptive learning. Nevertheless, the reduced recall of stealthy attacks is evidence of the requirement to have more powerful multi-turn and retrieval-aware detection systems.

7.7 Limitations of the Evaluation

The testing of the USOF prototype also has a number of significant drawbacks. The research made use of a relatively small data sample of just 200 samples. Though this enabled controlled experimentation, it restricts the extrapolability of the results. Also, ITTAM showed moderate recall of the more stealthy kinds of attacks, especially indirect prompt injections and multi-turn attacks. The recently introduced Model Integrity and Supply Chain Validator (MISCV) is based on simplified behavioral metrics and does not have advanced deep embeddings. Lastly, the prototype was not scaled to real Large Language Models and was not tested at scale in production-like setups, which limits the extrapolability of the results.

7.8 Threats to Validity and Future Work

There are a number of threats to validity that should be considered. The main issues are high dependency on data, the possibility of overfitting in the ITTAM model, and simplified implementations of some modules, especially MISCV. Such factors can have an impact on the extent to which the performance that is observed can be translated to real-world deployments.

Future research will be done to overcome these limitations by using a number of directions. They include the creation and use of larger real-world datasets with actual LLM interaction, the use of advanced sentence embeddings and correct statistical drift detection techniques (including Kolmogorov-Smirnov tests). Further work will include the implementation of the framework with the popular LLM orchestration tools, including LangChain, vLLM, and NVIDIA NeMo Guardrails. Additional adversarial robustness testing with state-of-the-art techniques (e.g., GCG attacks) and thorough real-time deployment experiments with latency benchmarking will also be done to reinforce the practical applicability of the framework.

8 Discussion

8.1 Comparative Advantages Over Existing Approaches

The USOF offers benefits over traditional security methods that take a structural approach instead of an incremental approach to security. The first benefit is that the USOF provides a complete lifecycle approach to security through AI interactions (from input to ingestion, to execution, and finally through output). Unlike point solutions, the USOF addresses all possible threats across the entire lifecycle of an AI interaction, eliminating the gaps that exploitative attackers can otherwise leverage to combine threats across different lifecycle stages. The second is that the USOF functions as a runtime control plane, which means the USOF can provide security to pre-trained third-party models where there may not have been any security applied during the training of the model. The third is that the MISCV uses behavioral fingerprinting to see model integrity issues, so the MISCV is able to verify models' integrity even when those models may not have their training data available, including closed-source or third-party models. The fourth is that the ALFMs' continuous learning method ensures that the framework will continue to improve with the identification of new attack patterns, thereby ensuring that the framework will remain current and will not experience negative changes from the use of the previous methods of the framework and require the use of static rules to have to manually reposition the many rules that were identified by the use of the framework.

8.2 Limitations and Constraints

The USOF approach has many weaknesses, which dictate under what circumstances it can be applied. To begin with, behavioral fingerprinting requires a reasonable degree of stability over a time span in order to form a reliable basis. Therefore, when frequent model tuning is observed, it must be carefully observed to avoid variations in detection effectiveness because of variation (Yao et al., 2024). Second, the multistage monitoring of session states increases the complexity of computation, which is proportional to the length of the interaction, which requires optimisation techniques to control the state and reduce latency in longer sessions (Derner et al., 2024). Although adaptive learning implementation would greatly enhance the ALFM module, there is also a risk that such a solution would also reveal the model to other possible threats, including attacks that can be made based on the manipulation of the training pipeline and worsen the detection process as time goes by. These threats are related to data poisoning and model supply chain attacks prevalent in machine learning pipelines (Goldblum et al., 2022; Wang et al., 2025). Secure access control mechanisms and anomaly detection mechanisms should therefore be utilised in the process of operation.

Similarly, the design of the USOF fails to address the inherent trade-off between security and usability of the access control system. Even though tighter configurations reduce the chances of security threats, they may increase the inability to use systems and make them less flexible. The trade-off between security and usability is a highly familiar term to the field of governance and AI systems cybersecurity (Derner et al., 2024; Yao et al., 2024). Therefore, domain knowledge is needed in determining optimal thresholds with respect to the trade-off between security and usability. The analysis of performance policies based on analysis by the ALFM can provide the data that can be used to establish appropriate thresholds. But reality is based on practical security experience in a particular field.

8.3 Future Research Directions

Future research in several areas would help to reinforce the theoretical basis of the framework. Formal (or mathematical) proofs of the security provided by a behavioral fingerprinting technique would provide greater assurance in making informed and appropriate deployment decisions; having these formal proofs is analogous to the evidence found from traditional cryptography for verifying the integrity of a message. In addition, newer and stronger retrieval security methods for RAGs, like the method developed by (Nandagopal, 2025) that uses a combination of pre-storing encrypted copies of RAG data and verifying the integrity of RAGs when retrieved, can become complementary pieces and work with ITTAM's provenance tagger. The use of many different entities (multi-tenancy) to share threat intelligence using federated learning raises many important questions about the ability to protect against adversarial attacks in federated systems, and this warrants further investigation, as it is an area of research that is lacking. Last, but not least, the extension of the enforcement of agency-based trust boundaries to heterogeneous multi-agent systems that include agents from different vendors with agency-based and differing agency-based trust levels is a very serious research challenge in the multi-agent security architecture area.

9 Conclusion

This paper introduces a comprehensive and integrated architectural framework called USOF, designed to provide a secure and resilient mechanism for protecting artificial intelligence (AI) and machine learning (ML) systems against multiple types of threats. This includes protection against potential attacks based on input, as well as ensuring that the model supply chain is not subject to compromise. As organisations continuously integrate AI systems into enterprise-level, high-stakes operational environments, it is becoming increasingly obvious that fragmented point-based security solutions have limitations; therefore, the USOF has been established to create a uniform security control plane, enforceable in real time and during run time, so that identical security is applied to the entire lifecycle of AI through all phases: ingestion of inputs, executing the model, and delivering output. The USOF has six closely integrated modules (or components) that work together in harmony: An Input Trust and Threat Assessment Module, A Context/Aware Policy (Enforcement) Engine, an Execution Isolation Layer, the Model Integrity and Supply Chain Validator, the RGE, and the Adaptive Learning and Feedback Mechanism module. All of these modules were designed to meet a specific aspect of AI security/take into account their own role, and work together within a common context-aware orchestration layer so that there is continuous visibility of threats and to create an environment for a coordinated response with the ability to adapt to improve across all stages of the provision of the system.

AI and Machine Learning Advances
Volume 2, Issue 2
ISSN: 3067-3216

This framework has many distinct innovations that set it apart from existing methodologies, such as the ability to detect multiple types of injection attacks through multiple vectors (direct, indirect, multi-turn and multimodal); behavioral fingerprinting techniques for validating model integrity based on training data; and dynamic, context-aware policies that adapt to changing risk levels or operational contexts. In addition, there are output-side governance mechanisms that go beyond traditional input-side protections, reducing the risk of compromising the safety of sensitive data through second-order attack dissemination. The framework's model-agnostic architecture is a further distinguishing characteristic. It supports a wide range of AI deployment configurations, from models developed in-house to those acquired from third-party vendors, and from cloud-hosted to on-premises and edge deployments. Crucially, it accommodates closed-source and third-party models for which training data and architectural details are not accessible.

Furthermore, the framework is designed for integration with existing enterprise security infrastructure, enabling AI security events to be incorporated into broader monitoring, auditing, and incident response workflows. As AI technologies continue to expand into high-stakes domains such as healthcare, finance, and critical infrastructure, the need for robust, lifecycle-spanning security architectures grows increasingly urgent. The USOF provides a systematic and practically grounded basis for meeting these challenges. Future work may focus on optimising system performance, reducing operational overhead, and extending the framework to emerging paradigms such as decentralised AI environments and heterogeneous autonomous multi-agent systems.

Conflict of Interest

The authors declare no conflict of interests

Funding

None

Acknowledgment

None

Data availability

No primary data were generated or analysed in this study. The framework is based on a synthesis of existing literature, all of which is cited within the manuscript.

References

Carlini, N., Tramer, F., Wallace, E., Jagielski, M., Herbert-Voss, A., Lee, K., Roberts, A., Brown, T., Song, D., & Erlingsson, U. (2021). Extracting training data from large language models. 30th USENIX security symposium (USENIX Security 21),

Chen, S., Zharmagambetov, A., Mahloujifar, S., Chaudhuri, K., Wagner, D., & Guo, C. (2025). Secalign: Defending against prompt injection with preference optimisation. Proceedings of the 2025 ACM SIGSAC Conference on Computer and Communications Security,

Das, B. C., Amini, M. H., & Wu, Y. (2025). Security and privacy challenges of large language models: A survey. ACM Computing Surveys, 57(6), 1-39.

Deng, Z., Guo, Y., Han, C., Ma, W., Xiong, J., Wen, S., & Xiang, Y. (2025). Ai agents under threat: A survey of key security challenges and future pathways. ACM Computing Surveys, 57(7), 1-36.

Derner, E., Batistič, K., Zahálka, J., & Babuška, R. (2024). A security risk taxonomy for prompt-based interaction with large language models. Ieee Access, 12, 126176-126187.

Dong, Y., Mu, R., Zhang, Y., Sun, S., Zhang, T., Wu, C., Jin, G., Qi, Y., Hu, J., & Meng, J. (2025). Safeguarding large language models: A survey. Artificial intelligence review, 58(12), 382.

Gabriel, I. (2020). Artificial Intelligence, Values, and Alignment: I. Gabriel. Minds and machines, 30(3), 411-437.

Goldblum, M., Tsipras, D., Xie, C., Chen, X., Schwarzschild, A., Song, D., Mądry, A., Li, B., & Goldstein, T. (2022). Dataset security for machine learning: Data poisoning, backdoor attacks, and defenses. IEEE Transactions on Pattern Analysis and Machine Intelligence, 45(2), 1563-1580.

Greshake, K., Abdelnabi, S., Mishra, S., Endres, C., Holz, T., & Fritz, M. (2023). Not what you've signed up for: Compromising real-world llm-integrated applications with indirect prompt injection. Proceedings of the 16th ACM workshop on artificial intelligence and security,

Gu, S. S., Lillicrap, T., Turner, R. E., Ghahramani, Z., Schölkopf, B., & Levine, S. (2017). Interpolated policy gradient: Merging on-policy and off-policy gradient estimation for deep reinforcement learning. Advances in Neural Information Processing Systems, 30.

Gu, T., Liu, K., Dolan-Gavitt, B., & Garg, S. (2019). Badnets: Evaluating backdooring attacks on deep neural networks. Ieee Access, 7, 47230-47244.

Gulyamov, S., Gulyamov, S., Rodionov, A., Khursanov, R., Mekhmonov, K., Babaev, D., & Rakhimjonov, A. (2026). Prompt Injection Attacks in Large Language Models and AI Agent Systems: A Comprehensive Review of Vulnerabilities, Attack Vectors, and Defense Mechanisms. Information, 17(1), 54.

Han, X., Zhang, Z., Ding, N., Gu, Y., Liu, X., Huo, Y., Qiu, J., Yao, Y., Zhang, A., & Zhang, L. (2021). Pre-trained models: Past, present and future. AI open, 2, 225-250.

Jiao, Y., Wang, X., & Yang, K. (2025). Pr-attack: Coordinated prompt-rag attacks on retrieval-augmented generation in large language models via bilevel optimisation. Proceedings of the 48th International ACM SIGIR Conference on Research and Development in Information Retrieval,

Kim, S., Yun, S., Lee, H., Gubri, M., Yoon, S., & Oh, S. J. (2023). Propile: Probing privacy leakage in large language models. Advances in Neural Information Processing Systems, 36, 20750-20762.

Kwon, H., & Pak, W. (2024). Text-based prompt injection attack using mathematical functions in modern large language models. Electronics, 13(24), 5008.

Mylrea, M., & Robinson, N. (2023). Artificial Intelligence (AI) trust framework and maturity model: applying an entropy lens to improve security, privacy, and ethical AI. Entropy, 25(10), 1429.

Nandagopal, S. (2025). Securing retrieval-augmented generation pipelines: A comprehensive framework. Journal of Computer Science and Technology Studies, 7(1), 17-29.

Shvetsova, O., Katalshov, D., & Lee, S.-K. (2025). Innovative Guardrails for Generative AI: Designing an Intelligent Filter for Safe and Responsible LLM Deployment. Applied Sciences, 15(13), 7298.

Wang, B., Yao, Y., Shan, S., Li, H., Viswanath, B., Zheng, H., & Zhao, B. Y. (2019). Neural cleanse: Identifying and mitigating backdoor attacks in neural networks. 2019 IEEE symposium on security and privacy (SP),

Wang, H., Guo, S., He, J., Liu, H., Zhang, T., & Xiang, T. (2025). Model supply chain poisoning: Backdooring pre-trained models via embedding indistinguishability. Proceedings of the ACM on Web Conference 2025,

Weidinger, L., Uesato, J., Rauh, M., Griffin, C., Huang, P.-S., Mellor, J., Glaese, A., Cheng, M., Balle, B., & Kasirzadeh, A. (2022). Taxonomy of risks posed by language models. Proceedings of the 2022 ACM conference on fairness, accountability, and transparency,

Yao, Y., Duan, J., Xu, K., Cai, Y., Sun, Z., & Zhang, Y. (2024). A survey on large language model (llm) security and privacy: The good, the bad, and the ugly. High-Confidence Computing, 4(2), 100211.

Yi, J., Xie, Y., Zhu, B., Kiciman, E., Sun, G., Xie, X., & Wu, F. (2025). Benchmarking and defending against indirect prompt injection attacks on large language models. Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining V. 1,