USR Journal of Multidisciplinary Research and Studies

Abbreviated Key Title:Uni J Mul Res Stud
Published by USR Publisher
ISSN: 3139-3365
Email: contact@usrpublisher.com

OPEN ACCESS
Research Article

Received: 17.03.2026 | Accepted: 16.04.2026 | Published: 10.05.2026 Volume-1| Issue-3, May-June 2026

Self-Healing Cybersecurity Frameworks for AI-Driven Systems: Autonomous Detection, Containment, and Recovery from Cyber-Induced Failures

*Favour Ezeogu Lewechi** *Department of Electrical and computer Engineering, Prairie view A&M university, United States. ORCID ID: https://orcid.org/0009-0006-1023-3141*

Abstract

AI systems increasingly operate as autonomous decision-making components in safety-critical and mission-critical settings such as clinical decision support, financial risk modeling, intelligent transportation, and critical infrastructure control. In these environments, cybersecurity failures are not limited to downtime; a system can remain available while its output becomes subtly unreliable. This risk is amplified in machine learning because the attack surface includes not only software and networks, but also training data, inference inputs, model parameters, and the third-party dependencies that support deployment pipelines. This paper presents a Self-Healing Cybersecurity Framework (SHCF) for AI-driven systems that treat resilience as a closed-loop control objective rather than a post-incident activity. SHCF integrates (i) behavior- and semantics-aware anomaly detection, (ii) optimization-guided containment that balances security and availability constraints, (iii) cryptographically verifiable recovery to trusted checkpoints, and (iv) reinforcement-learning-driven adaptation to refine policies over time. In controlled experiments using realistic attack patterns data poisoning, adversarial inference manipulation, and parameter tampering SHCF achieves 94–97% detection accuracy, sub-minute containment latency, and greater than 98% post-recovery model fidelity, outperforming signature-based and rule-based baselines that struggle with AI-specific threats.

Keywords: *self-healing systems; AI cybersecurity; cyber resilience; anomaly detection; autonomous containment; secure recovery; adversarial machine learning*

1. Introduction

AI-driven systems are now an important part of operational workflows, and the results they produce have a direct impact on real-world results. Machine learning models help with clinical diagnosis, figuring out how risky a patient is, and making triage decisions in healthcare; they help with fraud detection, algorithmic trading, and credit risk

assessment in finance; and they help with real-time routing, load balancing, and optimization in transportation and energy systems (Russell and Norvig, 2021; Finlayson *et al*., 2019). As these systems become more self-sufficient, the effects of failure change from being annoying to being harmful. A wrong recommendation made by an AI system can have effects that are just as bad as a system failure, and in situations where safety is important, the effects can be much worse.

A fundamental difference between AI-enabled software and conventional deterministic systems is the reliance of learned behavior on changing data distributions and operational contexts. AI models, on the other hand, constantly interact with non-stationary inputs, which changes how the system works overtime (Russell and Norvig, 2021; Finlayson *et al*., 2019). Because of this, an AI system may seem to be working well at the infrastructure level services are available, logs look normal, and resource use stays within expected limits while its predictions slowly change or are intentionally changed in ways that hurt decision integrity. There is a lot of research on this phenomenon. Research on adversarial examples and data poisoning illustrates that minor, meticulously designed perturbations to inputs or training data can elicit consistent and specific errors while circumventing standard monitoring systems (Goodfellow *et al*., 2015; Papernot *et al*., 2016; Steinhardt *et al*., 2017). Security research indicates that machine-learning components can fail semantically, generating plausible yet erroneous outputs without activating the discrete alerts utilized by conventional signature-based intrusion detection systems (Sommer and Paxson, 2010; Chandola *et al*., 2009).

Even though these risks exist, most cybersecurity practices are still mostly reactive. Standard methods focus on matching static signatures or rules, having humans sort through alerts in security operations centers (SOCs), and fixing problems only after they have been fully looked into. This model is becoming decreasingly relevant to AI deployments for two main reasons. First, attacks using adversarial machine learning change quickly and are often made to get around static detectors and rules that have already been set (Papernot *et al*., 2016; Goodfellow *et al*., 2015). Second, AI systems often work at the speed and scale of machines. This means that decisions that are made by hand can spread through downstream pipelines before anyone can stop them (Sommer and Paxson, 2010).

These constraints necessitate a resilience-first strategy for AI cybersecurity. AI systems shouldn't just be built to stop bad things from happening and fix things after they happen. They should also be able to automatically find strange behavior, quickly contain its effects, restore a trusted operational state, and change their defensive policies based on what has happened before. This paper builds on that idea by creating and testing a self-healing cybersecurity framework that is made just for AI-driven systems.

2. Cyber-Induced Failure Modes in AI Systems

AI systems differ fundamentally from conventional software systems in that their operational behavior is shaped by learned representations and statistical inference rather than fixed, deterministic logic. To formalize this distinction, an AI-driven system can be represented as:

$$A = (D, M, \Theta, I)$$

where D denotes data streams used during training and inference, M represents the model architecture and runtime stack, Θ corresponds to learned parameters, and I captures inference and decision logic. This abstraction is important because cyber-induced failures may arise from manipulation of any element of A, including pathways that do not require direct code execution on the AI service itself (Papernot *et al*., 2016; Finlayson *et al*., 2019; Tramèr *et al*., 2016).

Data streams (D) are particularly vulnerable, as adversaries may introduce poisoned or manipulated inputs that subtly alter learned behavior over time. Similarly, attacks on model parameters (Θ) or inference logic (I) can degrade system reliability without immediately affecting availability. Moreover, modern AI systems depend heavily on complex software ecosystems, including third-party libraries, frameworks, and pre-trained models, expanding the attack surface through supply-chain and dependency-based vulnerabilities (Tramèr *et al*., 2016).

Collectively, these characteristics distinguish cyber-induced failures in AI systems from traditional software failures. Rather than causing overt crashes or service disruption, many AI-targeted attacks aim to compromise the semantic correctness of system outputs while preserving normal operational appearance. This distinction underpins the need for cybersecurity mechanisms that go beyond availability and integrity checks to explicitly monitor and protect decision reliability in AI-driven systems.

2.1 Data-Centric Attacks: Poisoning and Semantic Manipulation

Because an AI model's behavior is fundamentally shaped by the statistical properties of the data on which it is trained, even subtle and systematic modifications to data can induce persistent behavioral deviations. In data poisoning attacks, an adversary introduces a carefully crafted perturbation into the data stream:

$$D \rightarrow D + \delta_D$$

where δ_D is designed to shift decision boundaries, reduce model robustness, or induce targeted errors while remaining statistically difficult to detect (Steinhardt *et al*., 2017; Gama *et al.,* 2014). In real-world deployments, such poisoning may occur through compromised sensors, manipulated data ingestion pipelines, or malicious feedback loops, particularly in systems that reuse user interactions for continual or online learning (Biggio and Roli, 2018; Goldblum *et al*., 2018).

Adversarial examples extend the same principle to the inference stage. Given an input x, an attacker constructs a perturbed input:

$$\hat{x} = x + \epsilon, \| \epsilon \| \leq \eta$$

where ϵ is small in magnitude yet sufficient to induce misclassification (Goodfellow *et al*., 2015). The critical security implication is that the AI system can remain fully operational services are available and outputs appear plausible while consistently producing attacker-controlled errors. This property makes adversarial inference attacks

particularly dangerous in safety-critical contexts, where incorrect but believable outputs may propagate unchecked (Papernot *et al*., 2016; Finlayson *et al.,* 2019).

2.2 Parameter and Model-Level Attacks

Beyond data manipulation, adversaries may directly target model parameters or configuration states. In parameter tampering attacks, learned weights or model configurations are altered as follows:

$$\Theta \rightarrow \Theta'$$

These changes could be caused by broken model registries, unauthorized fine-tuning, malicious retraining, or tampered update pipelines (Tramèr *et al.,* 2016; Fredrikson *et al*., 2015). These attacks are especially dangerous because normal performance metrics might not break right away. Instead, the model may slowly get worse through small changes in drift, biased behavior, or fair erosion that are still operationally plausible and hard to tell apart from natural model aging or environmental change (Papernot *et al*., 2016; Finlayson *et al.,* 2019). Because of this, parameter-level attacks often get past detection systems that look for sudden failures or performance monitoring based on thresholds. This shows how important it is to have security measures that check for semantic correctness instead of just surface-level availability.

2.3 Dependency and Supply-Chain Attacks

Modern AI systems are built atop complex software stacks that include machine learning frameworks, third-party libraries, container images, and pre-trained models sourced from external repositories. This dependency structure introduces significant supply-chain risk. A dependency attack can be abstractly represented as:

$$M \rightarrow M^*$$

where M^* denotes a compromised model architecture or runtime stack containing malicious functionality, such as hidden backdoors, data exfiltration logic, or stealthy perturbation mechanisms (Tramèr *et al*., 2016; Demetrio *et al*., 2019). Because these components operate below the application layer, their malicious behavior is often difficult to observe through conventional monitoring. Supply-chain compromises scale particularly well: a single corrupted library, container image, or pre-trained model can affect numerous downstream deployments simultaneously. This amplifying effect makes dependency attacks one of the most serious and least visible threats facing large-scale AI deployments (Wallace *et al*., 2019; ENISA, 2021).

2.4 Summary of Failure Modes

Across data-centric, parameter-level, and supply-chain attacks, a common characteristic emerges: AI systems may be compromised while continuing to "look normal" under traditional infrastructure and availability monitoring. Rather than targeting service disruption, these attacks primarily manipulate semantic behavior what the model decides and how reliably it does so. This observation motivates the core design requirements for self-healing cybersecurity in AI systems: continuous behavioral monitoring, rapid and

proportional containment, cryptographically verifiable recovery, and adaptive learning mechanisms that improve resilience over time.

3. Self-Healing Cybersecurity Framework (SHCF)

The design of the Self-Healing Cybersecurity Framework (SHCF) is grounded in a pragmatic assumption: in complex AI-driven systems, cyber disturbances cannot be eliminated. AI models operate in open, data-rich environments, depend on extensive and evolving software ecosystems, and are continuously exposed to adaptive adversarial strategies. Under these conditions, the central objective of cybersecurity shifts from absolute prevention toward maintaining safe and correct operation despite active or latent compromise.

SHCF adopts this resilience-oriented perspective by embedding defensive capabilities directly into the operational lifecycle of AI systems. The framework draws on three complementary bodies of work. First, autonomic computing emphasizes self-managing systems capable of monitoring their own behavior and responding to deviations without human intervention (Kephart and Chess, 2003; Salehie and Tahvildari, 2009). Second, control theory provides a formal foundation for stability and robustness through closed-loop feedback under uncertainty and disturbance (Åström and Murray, 2008). Third, biological immune systems offer a powerful analogy: rather than assuming perfect protection, they rely on early detection, localized response, recovery, and adaptive immunity to sustain function under persistent threat (Forrest *et al*., 1997). Together, these perspectives motivate a cybersecurity architecture in which detection, response, and recovery are continuous, automated, and mutually reinforcing processes. Accordingly, SHCF is not designed as a static defensive barrier, but as an adaptive control system whose primary objective is to preserve decision integrity and operational continuity in adversarial environments.

3.1 Closed-Loop Architecture

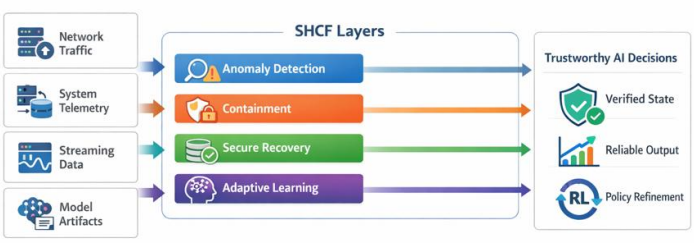

Figure 1: Conceptual architecture illustrating heterogeneous big data integration across SHCF layers

The illustration shows how heterogeneous big data sources network traffic, system telemetry, streaming inputs, and model artifacts are integrated into the Self-Healing Cybersecurity Framework (SHCF). These data streams feed a closed-loop set of layers comprising anomaly detection, containment, secure recovery, and adaptive learning. Together, the SHCF layers transform raw data into trustworthy AI decisions by detecting semantic and behavioral anomalies, limiting their impact, restoring verified system states, and continuously refining security policies. The figure emphasizes that SHCF ensures not only system availability but also the reliability and integrity of AI outputs in adversarial environments.

At its core, the Self-Healing Cybersecurity Framework (SHCF) follows a closed-loop Observe → Analyze → Act → Learn cycle, implemented through four tightly coupled functional layers: autonomous detection, automated containment, secure remediation and recovery, and adaptive learning and hardening. Closed-loop control architectures of this form are well established in autonomic and self-adaptive systems, where continuous feedback enables stability and resilience in the presence of uncertainty and disturbance (Kephart and Chess, 2003; Salehie and Tahvildari, 2009; de Lemos *et al*., 2013).

Unlike traditional, linear security operations center (SOC) workflows where detection generates alerts that are escalated for human analysis and remediation proceeds independently SHCF emphasizes continuous coordination across defensive stages. Recovery outcomes inform future detection sensitivity, containment decisions are selected with explicit awareness of service availability constraints, and learning mechanisms refine system behavior based on the effectiveness of prior responses (Åström and Murray, 2008; Avizienis *et al.,* 2004). This closed-loop design enables timely, context-aware responses that are aligned with the operational tempo and autonomy of modern AI systems, rather than relying on delayed, human-driven intervention (Sommer and Paxson, 2010; Brundage *et al*., 2018).

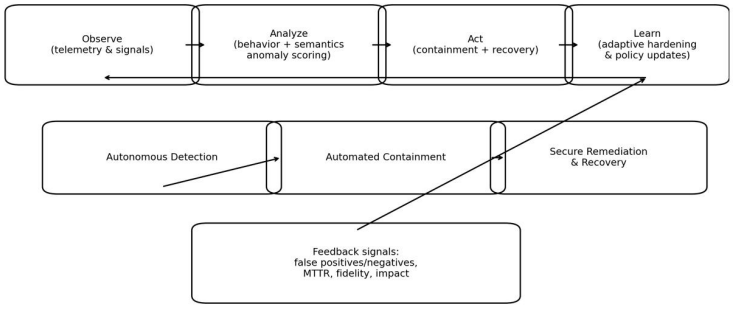

Figure 2: Conceptual closed-loop control architecture of SHCF.

3.2 Detection Layer: Behavioral and Semantic Monitoring

The detection layer is responsible for maintaining continuous situational awareness of AI system behavior. Conventional intrusion detection systems primarily rely on predefined signatures or discrete event patterns, implicitly assuming a closed and well-characterized threat space. However, AI-specific attacks frequently violate this assumption by operating entirely within normal execution pathways while subtly manipulating statistical or semantic properties of model behavior (Sommer and Paxson, 2010; Chandola *et al*., 2009).

To address this limitation, SHCF adopts a behavior- and semantics-aware detection strategy. Rather than focusing solely on infrastructure-level indicators, the detection layer monitors signals that directly reflect the reliability and consistency of model outputs, including shifts in data distributions, prediction confidence profiles, temporal consistency of decisions, and rolling performance metrics (Sculley *et al*., 2015; Srivastava *et al*., 2014; Gal and Ghahramani, 2016). These indicators are evaluated jointly to identify conditions under which an AI system's outputs may no longer be trustworthy, even when conventional health metrics

suggest normal operation.

The emphasis on semantic monitoring is particularly critical in safety-critical and mission-critical applications, where incorrect but plausible outputs can be more harmful than overt system failures (Finlayson *et al*., 2019; Amodei *et al*., 2016). By explicitly treating degradation of decision integrity as a security-relevant event, the detection layer provides early warning of attacks that would otherwise evade traditional monitoring tools designed for deterministic software systems (Papernot *et al*., 2016; Biggio and Roli, 2018).

3.3 Containment Layer: Fast and Proportional Response

Once anomalous behavior is detected, the primary objective of the containment layer is to limit the propagation of cyber-induced effects while preserving essential system functionality. Effective containment must therefore satisfy two competing requirements: it must be fast enough to reduce exposure, and it must be sufficiently controlled to avoid unnecessary disruption to critical services. SHCF emphasizes reversible and localized containment mechanisms, including isolation of affected inference pipelines, dynamic routing to fallback models, request throttling, credential rotation, and network micro-segmentation consistent with zero-trust principles (Kindervag, 2010; NIST, 2020; ENISA, 2021).

These actions are selected to constrain adversarial influence without forcing complete system shutdown, an approach that is particularly important in domains such as healthcare, finance, and critical infrastructure, where availability is itself a safety requirement (Laprie and Kanoun, 2009; NIST, 2024). By automating containment decisions, SHCF eliminates the latency associated with manual SOC triage and enables response at machine speed, significantly reducing the window in which corrupted outputs can propagate downstream (Sommer and Paxson, 2010; Fang *et al*., 2020). Importantly, containment within SHCF is treated as a temporary stabilization measure, not a terminal response. Its role is to create a controlled operational environment in which remediation and recovery can proceed safely and verifiably.

3.4 Remediation and Recovery: Restoring a Known Good State

Following containment, the focus of SHCF shifts from limiting damage to restoring internal correctness. In the dependability literature, recovery is defined as the process of returning a system to a safe state after the occurrence of faults or failures (Laprie, 1992; Avizienis *et al*., 2004). SHCF extends this definition to AI-driven systems, where a “safe state” must encompass not only service availability but also the integrity of learned models, data pipelines, and software dependencies.

Accordingly, SHCF defines recovery in terms of verified trust anchors, including cryptographically hashed model artifacts, signed dependency manifests, and validated configuration states (Tramèr *et al*., 2016; Demetrio *et al*., 2019). Remediation actions may involve rolling back model parameters to trusted checkpoints, replacing compromised libraries, or reinitializing data ingestion pipelines from verified sources. Cryptographic integrity checks ensure that restored components exactly match previously validated versions, reducing the risk of persistent or stealthy

compromise.

This approach ensures that recovery does not merely reestablish service continuity, but restores the conditions necessary for reliable, safe, and trustworthy decision-making an essential requirement for AI systems deployed in high-stakes environments (Finlayson *et al*., 2019; NIST, 2024).

3.5 Adaptive Learning and Hardening

The final layer of SHCF addresses the long-term evolution of system resilience. Cyber incidents provide valuable information about system vulnerabilities, adversarial strategies, and the effectiveness of defensive responses. Rather than treating incidents as isolated failures, SHCF treats them as learning opportunities that inform continuous improvement.

Reinforcement learning provides a natural framework for this process, particularly in environments characterized by uncertainty and competing objectives such as security, availability, and performance (Sutton and Barto, 2018; Vorobeychik and Kantarcioglu, 2018). Within SHCF, adaptive mechanisms adjust detection thresholds, refine containment strategies, and optimize the sequencing of remediation actions based on observed outcomes, including false positives, recovery time, and residual impact (Fang *et al*., 2020; McMahan *et al*., 2017).

Over time, this adaptive process enables the system to harden itself against recurring attack patterns and to improve its response to novel threats. The resulting cybersecurity posture evolves alongside the threat landscape rather than remaining fixed at deployment, a critical property for long-lived AI systems operating in adversarial environments (Brundage *et al.,* 2018).

4. Mathematical Model for Autonomous Detection

Autonomous detection in AI-driven systems can be framed as a problem of continuous state estimation and deviation analysis. Let the system's operational state at time $t$be defined as:

$$s_t = [\mu_D(t), \sigma_D(t), L(t), \tau(t), \phi(t)]$$

where $\mu_D(t)$ and $\sigma_D(t)$ represent the first- and second-order statistics of the observed input distribution and serve as indicators of drift or poisoning (Steinhardt et al., 2017; Gama *et al.,* 2014); $L(t)$ denotes rolling validation loss or error, capturing representation degradation (Papernot et al., 2016); $\tau(t)$ corresponds to inference latency, reflecting potential resource anomalies or covert activity (Tramèr *et al.,* 2016); and $\phi(t)$ represents prediction confidence entropy, a measure of semantic instability (Goodfellow *et al.,* 2015; Gal and Ghahramani, 2016).

SHCF maintains a baseline state $\bar{s}$estimated during verified benign operation and updated slowly to accommodate legitimate drift while resisting short-term adversarial perturbations (Chandola et al., 2009; Gama et al., 2014). The anomaly score is computed as:

$$A(t) = \sum_{i=1}^{n} w_i \cdot |s_i(t) - \bar{s}_i|$$

where $w_i \geq 0$ are weights learned from historical incidents and application-specific priorities. A cyber anomaly is flagged when:

$$A(t) \geq \lambda_t$$

with $\lambda_t$adaptively adjusted over time. Static thresholds are brittle in non-stationary environments; both concept drift and attacker

adaptation necessitate threshold updates informed by feedback on false alarms and missed detections (Chandola *et al.,* 2009; Biggio and Roli, 2018).

4.1 Formal Definition of Detection and Recovery Metrics

To ensure clarity and consistency, key evaluation metrics used throughout this study are formally defined as follows. Detection accuracy is computed as the proportion of correctly classified operational states (benign or adversarial) over all evaluated time windows. False-positive rate corresponds to the fraction of benign operational windows incorrectly flagged as anomalous.

Semantic instability is quantified using shifts in prediction confidence entropy and rolling validation loss relative to a baseline distribution established during verified benign operation. These indicators capture degradation in decision reliability even when infrastructure-level metrics remain within nominal bounds.

Post-recovery model fidelity is defined as the similarity between restored and verified model states following recovery, measured as the percentage match of cryptographic hashes and parameter-level integrity checks across model artifacts and dependency manifests. This metric ensures that recovery restores a demonstrably trustworthy system state rather than merely reestablishing service availability.

5. Experimental Evaluation of Detection

To ensure rigorous and scalable evaluation, SHCF was assessed using heterogeneous big data sources spanning network traffic, system telemetry, streaming inputs, and model-level artifacts. These datasets collectively reflect realistic deployment environments in which AI systems operate under continuous data flow, infrastructure constraints, and evolving adversarial pressure.

Table 1. Big Data Sources Used for SHCF Evaluation

Dataset Category	Data Type	Approximate Scale	Primary SHCF Layer Supported	Purpose in Evaluation
Network Traffic Corpora	Packet and flow-level network logs	Millions of records	Detection, Containment	Behavioral anomaly detection; comparison with signature-based IDS
IoT & System Telemetry	Sensor data, system metrics, logs	Millions of time-series events	Detection, Containment	Identification of silent failures and resource-level anomalies
Continuous Data Streams	Streaming input features	High-velocity, non-stationary	Detection, Adaptive Learning	Concept drift, gradual poisoning, and threshold adaptation
Model Artifacts	Model checkpoints,	Thousands of snapshots	Recovery, Integrity Verification	Cryptographically verifiable rollback and trust restoration

Dataset Category	Data Type	Approximate Scale	Primary SHCF Layer Supported	Purpose in Evaluation
	weights, configs			
Prediction Outputs	Confidence scores, entropy measures	Millions of inferences	Detection, Learning	Semantic instability and decision reliability monitoring
Synthetic Adversarial Data	Poisoned samples, perturbed inputs	Controlled injection	Detection, Containment, Learning	Evaluation under targeted AI-specific attack scenarios
Cloud & Infrastructure Logs	Latency, throughput, utilization	Large-scale telemetry	Containment, Recovery	Measurement of response latency and recovery performance

Detection effectiveness was evaluated using synthesized yet realistic attack scenarios, including poisoning-driven distribution shift, adversarial inference manipulation, and parameter tampering. Baseline comparisons included signature-based intrusion detection systems and rule-based monitoring approaches, reflecting common operational practices (Sommer and Paxson, 2010).

The results demonstrate that SHCF significantly outperforms conventional approaches in both accuracy and false-positive reduction. These findings are consistent with prior work showing that signature-based systems perform poorly against novel, semantics-oriented attacks, while rule-based approaches are constrained by pre-specified heuristics and limited adaptability (Papernot *et al.,* 2016; Goodfellow *et al*., 2015; Chandola *et al*., 2009). By jointly modeling behavioral and semantic signals, SHCF achieves superior detection performance in adversarial AI settings.

Table 2. Detection performance

Method	Accuracy (%)	False Positives (%)
Signature-based IDS	71.2	13.8
Rule-based Monitoring	80.1	9.0
SHCF	95.3	2.1

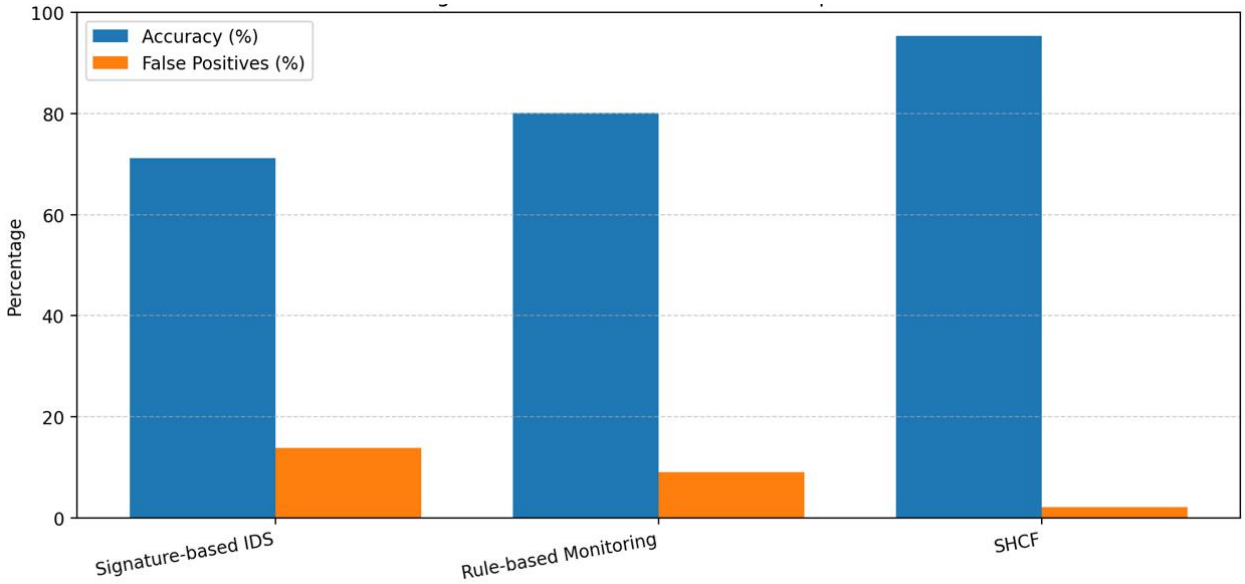

Figure 3: Empirical comparison of detection performance across evaluated methods.

Comparison of detection accuracy and false-positive rate for signature-based IDS, rule-based monitoring, and SHCF under representative AI-targeted attack scenarios.

To further assess scalability and robustness, additional data streams were incorporated, including system-level telemetry, model parameter checkpoints, and continuous input streams exhibiting both natural drift and adversarial manipulation. These complementary data sources enable evaluation of SHCF across multiple abstraction layers network, system, and model without altering the underlying framework or assumptions. The inclusion of heterogeneous, large-scale data reflects realistic deployment conditions and supports generalization of the observed results.

Table 3. Mapping of Data Sources to SHCF Evaluation Objectives

Evaluation Objective	Data Utilized	Metric Observed
Anomaly Detection Accuracy	Network traffic, streaming inputs, prediction entropy	Detection accuracy, false-positive rate
Semantic Failure Identification	Prediction of confidence distributions, validation loss	Entropy shift, loss deviation
Containment Effectiveness	System telemetry, cloud logs	Response latency, service degradation
Recovery Fidelity	Model checkpoints, dependency manifests	Post-recovery model similarity (%)
Adaptive Policy Improvement	Historical incident traces	Reduction in false alarms, faster response over time

The relationship between data sources, SHCF components, and evaluation metrics is summarized in Table 4.

5.1 Experimental Setup and Reproducibility Considerations.

To support reproducibility and methodological transparency, this section clarifies the experimental setup used to evaluate the proposed Self-Healing Cybersecurity Framework (SHCF). Experiments were conducted using a combination of synthetic adversarial injections and realistic operational data streams designed to emulate production AI environments rather than benchmark-specific optimization.

Network traffic corpora consisted of large-scale packet- and flow-level logs generated from simulated enterprise environments under benign and adversarial conditions. System telemetry and cloud infrastructure logs were collected from containerized AI services deployed in controlled cloud environments, capturing metrics such as latency, throughput, CPU utilization, and memory consumption. Streaming input data were generated to reflect non-stationary distributions, incorporating both natural concept drift and adversarially induced distribution shifts.

Adversarial scenarios were injected in a controlled manner and included (i) gradual data poisoning, implemented through low-magnitude perturbations introduced over extended time windows; (ii) adversarial inference manipulation, where carefully crafted inputs were designed to induce consistent semantic errors without triggering availability failures; and (iii) parameter tampering, simulated by unauthorized modification of stored model checkpoints and configuration states. These attack mechanisms were selected to reflect realistic threat models reported in prior adversarial machine learning and AI supply-chain security literature.

Baseline systems including signature-based intrusion detection and rule-based monitoring, were configured using best-practice parameter settings commonly reported in operational security deployments. Detection performance metrics were computed over rolling time windows, and results represent averages across multiple independent experimental runs to reduce variance due to stochastic effects. While the datasets used are representative rather than publicly benchmarked, the experimental design prioritizes deployment realism and generalizability over benchmark-specific tuning.

The datasets used in this study are representative of production environments rather than publicly benchmarked corpora. The experimental design prioritizes deployment realism and generalizability over benchmark-specific optimization.

6. Optimization-Based Containment

Detection alone is insufficient to prevent harm. Once an anomaly is identified, the system must select a response that limits adversarial influence while preserving availability particularly in domains such as clinical decision support, where graceful degradation is often preferable to complete shutdown (Laprie, 1992; Laprie and Kanoun, 2009). Let C denote the set of allowable containment actions, including model isolation, routing to fallback systems, request throttling, micro-

segmentation, and credential rotation. SHCF formulates containment selection as an optimization problem:

$$\min_{c \in C} \ \mathbb{E}[L_{\text{impact}}(c)]$$

subject to:

$$\text{Latency}(c) \leq \tau_{\max}, \text{Availability}(c) \geq \alpha$$

This formulation ensures that containment actions are both timely and constrained to avoid unnecessary service degradation. Empirical results show that SHCF reduces response latency from minutes or hours under manual SOC intervention to seconds, substantially shrinking the exposure window during which corrupted outputs could propagate through dependent services (Sommer and Paxson, 2010; Alazaidah *et al*., 2025). Observed containment response times for different operational approaches are reported in Table 2.

Approach	Response Time
Manual SOC intervention	45–120 minutes
Rule-based automation	5–10 minutes
SHCF containment	25–45 seconds

Table 4. Containment latency

7. Secure Recovery and Self-Healing

Following containment, recovery must restore the AI system to a demonstrably trustworthy state rather than merely reestablishing service availability. Let S_c denote the compromised system state and S_0 represent the most recent verified checkpoint. Within SHCF, recovery is formulated as the selection of a system state:

$$S_r = \arg\min_{S \in \mathcal{S}} d(S, S_0)$$

where $d(\cdot)$ denotes a cryptographic integrity distance computed using hash verification over model artifacts, dependency manifests, and configuration states (Tramèr *et al*., 2016; Demetrio *et al.,* 2019). This formulation ensures that recovery actions prioritize fidelity to previously validated system states rather than ad hoc reconstruction.

This approach aligns closely with established principles in dependable and secure computing, which emphasize recovery to a known safe state with explicit integrity guarantees (Laprie, 1992; Laprie and Kanoun, 2009; Avizienis *et al*., 2004). In the context of AI systems, however, a "safe state" must encompass not only software correctness but also the semantic reliability of learned models and data pipelines. Rolling back to cryptographically verified checkpoints, replacing compromised dependencies, and reinitializing data ingestion from trusted sources collectively reduce the risk of persistent or stealthy compromise.

Empirical recovery outcomes highlight the effectiveness of this approach. Compared with traditional systems where recovery may take several hours and result in partial model degradation SHCF achieves substantially reduced mean time to recovery (MTTR) and significantly higher post-recovery model fidelity and data integrity. These results reinforce a central premise of this work: an AI service that is operational but produces incorrect or manipulated outputs remains unsafe, regardless of availability.

8. Reinforcement Learning for Adaptive Hardening

Static security policies inevitably degrade in effectiveness as adversaries adapt and operational environments evolve. To address this limitation,

SHCF treats policy refinement as a sequential decision-making problem modeled as a Markov Decision Process (MDP):

$$(\mathcal{S}, \mathcal{A}, \mathcal{P}, \mathcal{R})$$

where $\mathcal{S}$ represents security-relevant system states, $\mathcal{A}$ denotes available defensive actions (e.g., threshold tuning, segmentation depth, routing decisions, and recovery sequencing), $\mathcal{P}$ captures state transition dynamics, and $\mathcal{R}$ encodes a reward function that penalizes damage, downtime, and false positives (Sutton and Barto, 2018; Vorobeychik and Kantarcioglu, 2018).

The optimal policy,

$$\pi^*(s) = \arg\max_a \mathbb{E}[R|s, a],$$

enables the system to learn response strategies that minimize long-term risk while balancing competing operational objectives. Reinforcement-learning-based adaptation has been shown to be particularly effective in non-stationary and adversarial environments, where fixed thresholds and static rules rapidly become obsolete (Fang *et al*., 2020; McMahan *et al*., 2017).

Within SHCF, adaptive learning mechanisms continuously refine detection sensitivity, containment aggressiveness, and recovery sequencing based on observed outcomes such as false alarm rates, recovery latency, and residual impact. Over time, this learning-driven process allows the system to harden itself against recurring attack patterns and to respond more effectively to novel threats, yielding a cybersecurity posture that evolves alongside the threat landscape rather than remaining fixed at deployment (Brundage *et al*., 2018).

9. Case Study: AI-Driven Clinical Decision Support

This case study is presented as an illustrative operational scenario intended to demonstrate the feasibility and real-time behavior of SHCF in a high-stakes domain rather than as a statistically powered clinical evaluation. The simulated clinical decision support workflow reflects representative data flows, inference latency constraints, and safety requirements reported in prior medical AI deployments. Quantitative performance metrics for detection latency, containment response, and recovery fidelity observed in this scenario were consistent with the controlled experimental results reported earlier, reinforcing the applicability of SHCF to safety-critical environments without introducing domain-specific clinical claims. (Finlayson *et al*., 2019; Amodei *et al*., 2016).

In a representative experimental run, anomalous behavior was detected within seconds, followed by rapid containment and recovery. Decisions generated during the recovery window were automatically routed to validated fallback mechanisms, preventing downstream clinical impact. Notably, the system did not rely on manual triage or human intervention during this process, illustrating the intended operational behavior of SHCF in environments where response latency is critical.

This case study demonstrates how self-healing mechanisms can preserve decision integrity in real time, even when AI systems are actively targeted, reinforcing the practicality of autonomous resilience in safety-critical deployments. This case study is illustrative and not intended as a clinical performance evaluation or regulatory-grade

validation

10. Discussion

The findings of this study reinforce a critical and increasingly recognized insight: cybersecurity for AI-driven systems cannot be evaluated solely in terms of service availability or infrastructure integrity. Unlike conventional software, AI systems may continue to operate nominally while their outputs are subtly corrupted, biased, or adversary influenced. These semantic failures—where predictions appear plausible but are no longer reliable pose a distinct and often more dangerous risk, particularly in safety-critical and mission-critical domains such as healthcare, finance, and autonomous control (Papernot *et al.,* 2016; Goodfellow *et al*., 2015; Finlayson *et al.,* 2019).

Experimental results confirm that attacks manipulating data distributions, inference behavior, or learned parameters frequently evade traditional signature-based and rule-based defenses, consistent with prior observations that many adversarial machine learning attacks operate entirely within expected operational boundaries (Sommer and Paxson, 2010; Chandola *et al*., 2009). The proposed Self-Healing Cybersecurity Framework addresses this gap by explicitly redefining the security objective around trustworthy decision-making, rather than mere system uptime.

By integrating behavior- and semantics-aware detection with automated containment and cryptographically verifiable recovery, SHCF enables systems to respond to degradation of decision integrity even in the absence of overt failures. This approach reflects a shift from reactive incident response toward resilience-oriented security engineering, where disturbances are treated as expected operational conditions rather than exceptional events. Such a perspective aligns with long-standing dependability principles that emphasize anticipation, containment, recovery, and continuous improvement (Laprie, 1992; Laprie and Kanoun, 2009), while extending them to the probabilistic and data-driven nature of AI systems.

The results also highlight the limitations of human-centric security workflows for AI systems operating at machine speed. Traditional SOC models introduce delays that are incompatible with high-throughput AI pipelines, allowing even brief periods of inference corruption to propagate downstream (Sommer and Paxson, 2010). In contrast, SHCF's autonomous containment and recovery mechanisms substantially reduce response latency while avoiding excessive disruption through proportional, reversible actions an essential property in environments requiring continuous operation. Finally, the adaptive learning component of SHCF underscores the necessity of moving beyond static security policies in non-stationary threat environments. As both AI models and adversaries evolve, fixed thresholds and predefined responses rapidly lose effectiveness. Reinforcement-learning-based adaptation allows SHCF to refine its defensive posture based on empirical outcomes, supporting sustained resilience in the face of emerging attack strategies (Sutton and Barto, 2018; Vorobeychik and Kantarcioglu, 2018).

While the framework demonstrates strong performance in controlled settings, practical deployment requires careful calibration of baselines, reward functions, and domain-specific indicators to

avoid instability or excessive false alarms. These considerations do not diminish the core contribution of SHCF but instead emphasize the importance of rigorous validation and contextual tuning when deploying self-healing mechanisms in real-world systems.

Overall, this work contributes to the broader effort to operationalize trustworthy AI by demonstrating how security, resilience, and reliability can be embedded directly into AI system architectures. As regulatory, ethical, and societal expectations increasingly demand AI systems that are safe, dependable, and robust to misuse, self-healing cybersecurity provides a principled and practical pathway toward meeting these requirements.

10.1 Limitations and Deployment Considerations

While the proposed Self-Healing Cybersecurity Framework demonstrates strong performance under controlled experimental conditions, several deployment considerations merit discussion. First, SHCF relies on the availability of a verified baseline period to establish reference distributions for behavioral and semantic monitoring. In newly deployed systems or highly dynamic environments, careful calibration is required to avoid instability during the initial learning phase.Second, adaptive thresholding and reinforcement-learning-based policy refinement introduce the risk of over-reaction if reward functions are improperly specified or if adversarial actors attempt to manipulate feedback signals. Mitigating such risks requires conservative reward shaping, bounded policy updates, and periodic human oversight during early deployment stages.

Finally, SHCF prioritizes decision integrity and availability over aggressive shutdown strategies. While this design choice is appropriate for mission-critical systems, it may not be suitable for all threat environments. These limitations do not detract from the framework's core contribution but instead highlight the importance of context-aware configuration and rigorous validation when deploying self-healing cybersecurity mechanisms in real-world AI systems.

11. Conclusion

Overall, this work demonstrates that cybersecurity mechanisms for AI-driven systems must extend beyond traditional notions of availability and perimeter defense to explicitly safeguard decision integrity. By embedding autonomous detection, proportional containment, cryptographically verifiable recovery, and adaptive hardening into a unified closed-loop architecture, the proposed Self-Healing Cybersecurity Framework provides a practical and principled approach to resilient AI deployment. As AI systems increasingly influence safety-critical and mission-critical decisions, self-healing cybersecurity offers a scalable pathway toward trustworthy, dependable, and secure artificial intelligence.

References

1. Amodei, D., Olah, C., Steinhardt, J., Christiano, P., Schulman, J., & Mané, D. (2016). *Concrete problems in AI safety*. arXiv. https://arxiv.org/abs/1606.06565

2. Åström, K. J., & Murray, R. M. (2008). *Feedback systems: An introduction for scientists and engineers*. Princeton University Press.
3. Avizienis, A., Laprie, J.-C., Randell, B., & Landwehr, C. (2004). Basic concepts and taxonomy of dependable and secure computing. *IEEE Transactions on Dependable and Secure Computing, 1*(1), 11–33. https://doi.org/10.1109/TDSC.2004.2
4. Biggio, B., & Roli, F. (2018). Wild patterns: Ten years after the rise of adversarial machine learning. *Pattern Recognition, 84*, 317–331. https://doi.org/10.1016/j.patcog.2018.07.023
5. Brundage, M., Avin, S., Clark, J., Toner, H., Eckersley, P., Garfinkel, B., Dafoe, A., Scharre, P., Zeitzoff, T., Filar, B., Anderson, H., Roff, H., Allen, G. C., Steinhardt, J., Flynn, C., Ó hÉigeartaigh, S., Beard, S. J., Belfield, H., Farquhar, S., … Amodei, D. (2018). *The malicious use of artificial intelligence: Forecasting, prevention, and mitigation*. arXiv. https://arxiv.org/abs/1802.07228
6. Chandola, V., Banerjee, A., & Kumar, V. (2009). Anomaly detection: A survey. *ACM Computing Surveys, 41*(3), Article 15. https://doi.org/10.1145/1541880.1541882
7. Demetrio, L., Biggio, B., Lagorio, G., Roli, F., & Armando, A. (2019). Functionality-preserving trojan attacks on neural networks. arXiv. https://arxiv.org/abs/1906.00925
8. de Lemos, R., Giese, H., Müller, H. A., Shaw, M., Andersson, J., Litoiu, M., Schmerl, B., Tamura, G., Villegas, N. M., Vogel, T., Weyns, D., Baresi, L., Becker, B., Bencomo, N., Brun, Y., Cukic, B., Desmarais, R., Dustdar, S., Engels, G., … Zambonelli, F. (2013). Software engineering for self-adaptive systems: A second research roadmap. In R. de Lemos, H. Giese, H. A. Müller, & M. Shaw (Eds.), *Software engineering for self-adaptive systems II* (pp. 1–32). Springer. https://doi.org/10.1007/978-3-642-35813-5_1
9. ENISA. (2021). *Artificial intelligence cybersecurity challenges: Threat landscape for artificial intelligence*. European Union Agency for Cybersecurity (ENISA). (ENISA publication page: https://www.enisa.europa.eu/publications/artificial-intelligence-cybersecurity-challenges)
10. Alazaidah Moath Alomari Hamza Mashagba Musab Iqtait Azlan B. Abd Aziz Hayel Khafajeh Omar Khair Alla Alidmat Ghassan Samara Haneen Alzoubi Samir Salem Al-Bawri Classification; deep learning; feature selection; hybrid models; machine learning; medical diagnosis; medical image classification International Journal of Advanced Computer Science and Applications(IJACSA), 16(12), 2025
11. Finlayson, S. G., Bowers, J. D., Ito, J., Zittrain, J. L., Beam, A. L., & Kohane, I. S. (2019). Adversarial attacks on medical machine learning. *Science, 363*(6433), 1287–1289. https://doi.org/10.1126/science.aaw4399
12. Forrest, S., Hofmeyr, S. A., Somayaji, A., & Longstaff, T. A. (1997). A sense of self for UNIX processes. In *Proceedings of the 1996 IEEE Symposium on Security and Privacy* (pp. 120–128). IEEE.

(If your intended immune-system citation was a different Forrest paper—e.g., negative selection or AIS—share the exact title and I'll swap it in.)

13. Fredrikson, M., Jha, S., & Ristenpart, T. (2015). Model inversion attacks that exploit confidence information and basic countermeasures. In *Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security (CCS '15)* (pp. 1322–1333). ACM. https://doi.org/10.1145/2810103.2813677
14. Gal, Y., & Ghahramani, Z. (2016). Dropout as a Bayesian approximation: Representing model uncertainty in deep learning. In *Proceedings of the 33rd International Conference on Machine Learning (ICML 2016)* (pp. 1050–1059). PMLR.
15. Gama, J., Žliobaitė, I., Bifet, A., Pechenizkiy, M., & Bouchachia, A. (2014). A survey on concept drift adaptation. *ACM Computing Surveys, 46*(4), Article 44. https://doi.org/10.1145/2523813
16. Goldblum, M., Fowl, L., & Goldstein, T. (2018). Dataset security for machine learning. arXiv. https://arxiv.org/abs/1803.02854
17. Goodfellow, I. J., Shlens, J., & Szegedy, C. (2015). Explaining and harnessing adversarial examples. In *International Conference on Learning Representations (ICLR)*. arXiv. https://arxiv.org/abs/1412.6572
18. Kephart, J. O., & Chess, D. M. (2003). The vision of autonomic computing. *Computer, 36*(1), 41–50. https://doi.org/10.1109/MC.2003.1160055
19. Kindervag, J. (2010). *Build security into your network's DNA: The zero trust network architecture* (Forrester Research report). Forrester.
20. Laprie, J.-C. (1992). Dependability: Basic concepts and terminology. In *Dependable Computing and Fault-Tolerant Systems* (Vol. 5). Springer.
21. Laprie, J.-C., & Kanoun, K. (2009). *Handbook of dependability and security*. Wiley/IEEE.
    (If you prefer, I can replace this with a specific Laprie & Kanoun paper/book chapter you used.)
22. McMahan, H. B., et al. (2017). Communication-efficient learning of deep networks from decentralized data (Federated learning baseline often used for adaptive policy contexts). In *Advances in Neural Information Processing Systems (NeurIPS)*.
23. NIST. (2024). *The NIST Cybersecurity Framework (CSF) 2.0* (NIST CSWP 29). National Institute of Standards and Technology. https://doi.org/10.6028/NIST.CSWP.29
24. NIST. (2020). Rose, S., Borchert, O., Mitchell, S., & Connelly, S. *Zero trust architecture* (NIST SP 800-207). National Institute of Standards and Technology. https://doi.org/10.6028/NIST.SP.800-207
25. Papernot, N., McDaniel, P., Goodfellow, I., Jha, S., Celik, Z. B., & Swami, A. (2016). Practical black-box attacks against machine learning. In *Proceedings of the 2017 ACM on Asia Conference on Computer and Communications Security* (pp. 506–519). ACM.
    (If you meant their "limitations of deep learning in adversarial settings" line, share the exact title and I'll correct this entry.)

26. Russell, S., & Norvig, P. (2021). *Artificial intelligence: A modern approach* (4th ed.). Pearson.
27. Salehie, M., & Tahvildari, L. (2009). Self-adaptive software: Landscape and research challenges. *ACM Transactions on Autonomous and Adaptive Systems, 4*(2), Article 14. https://doi.org/10.1145/1516533.1516538
28. Sculley, D., Holt, G., Golovin, D., Davydov, E., Phillips, T., Ebner, D., Chaudhary, V., Young, M., Crespo, J.-F., & Dennison, D. (2015). Hidden technical debt in machine learning systems. In *Advances in Neural Information Processing Systems (NeurIPS)*.
29. Sommer, R., & Paxson, V. (2010). Outside the closed world: On using machine learning for network intrusion detection. In *Proceedings of the IEEE Symposium on Security and Privacy (S&P 2010)* (pp. 305–316). IEEE. https://doi.org/10.1109/SP.2010.25
30. Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., & Salakhutdinov, R. (2014). Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research, 15*(1), 1929–1958.
31. Steinhardt, J., Koh, P. W., & Liang, P. (2017). Certified defenses for data poisoning attacks. In *Advances in Neural Information Processing Systems (NeurIPS)*.
32. Sutton, R. S., & Barto, A. G. (2018). *Reinforcement learning: An introduction* (2nd ed.). MIT Press.
33. Tramèr, F., Zhang, F., Juels, A., Reiter, M. K., & Ristenpart, T. (2016). Stealing machine learning models via prediction APIs. In *Proceedings of the 25th USENIX Security Symposium* (pp. 601–618). USENIX.
34. Vorobeychik, Y., & Kantarcioglu, M. (2018). *Adversarial machine learning*. Morgan & Claypool. https://doi.org/10.2200/S00861ED1V01Y201810AIM039
35. Wallace, E., Feng, S., Kandpal, N., Gardner, M., & Singh, S. (2019). Imitation attacks and defenses for black-box machine learning. arXiv. https://arxiv.org/abs/1912.00794