International Journal of Creative and Open Research in Engineering and Management
ISSN: **3108-1754 (Online)**
Volume 02 Issue 05 May-2026 | Impact Factor: 3.5

# Aegis-Edge: Gated Agentic Cyber Defense with Deception-Oriented Control for IoT/IIoT Edge Systems

**Anjali Srivastava ,**

Department of Computer Science and Engineering, J.C Bose University of Science and Technology,

YMCA ,Faridabad,India Email: miss.srivastavaa@gmail.com

***Abstract*—**Agentic AI is shifting cybersecurity from passive, reactive classification toward systems that reason, plan, and act across extended operational loops. Yet existing defenses remain constrained by high computational cost, excessive response la-tency, and unresolved questions of safe autonomy, particularly in IoT and Industrial IoT (IIoT) edge environments. This paper proposes *Aegis-Edge*, a novel agentic cyber-defense frame-work that integrates lightweight anomaly sensing, uncertainty-triggered reasoning, and deception-aware response orchestration specifically designed for resource-constrained edge deployments. The framework is motivated by a clear gap in the literature: while AI-for-cybersecurity surveys outline the need for advanced methods and new infrastructures, and human-in-the-loop XAI studies emphasize transparent decision support, neither provides a closed-loop autonomous architecture for the edge. Aegis-Edge addresses this gap through a gated four-agent pipeline: a compact *Monitor Agent* for always-on anomaly scoring; a *Reasoner Agent* activated only under uncertainty, concept drift, or escalating attack severity; a *Deception Agent* that selects bounded response primitives including canary tokens, honeypot redirec-tion, and decoy credential injection; and a *Governor Agent* that enforces policy constraints and human-approval thresholds. We describe the system architecture, formal optimization objective, experimental protocol, and an illustrative simulation study over standard public intrusion datasets. Simulated results indicate that the gated design reduces end-to-end response latency and energy consumption substantially relative to monolithic agentic baselines, while maintaining recall above 97.8% and reducing false positive rate to 1.1%. These findings position Aegis-Edge as a credible foundation for operationally viable, safe-by-design autonomous cyber defense at the edge.

***Index Terms*—agentic AI, cyber deception, edge security, intrusion detection, IoT/IIoT, resource-constrained systems, multi-agent systems, autonomous response**

## I. Introduction

Cybersecurity operations are increasingly defined by three intersecting pressures: the speed at which adversaries adapt their tactics, the scale of modern attack surfaces spanning billions of connected devices, and the operational constraints imposed by edge environments where compute, memory, and power are strictly bounded. Classical machine-learning defenders address part of this challenge by classifying traffic patterns with high statistical accuracy, but they rarely reason over multi-step response sequences, adapt their policies from feedback, or coordinate defensive actions across extended time horizons [1]. Agentic AI systems, which combine reasoning, planning, tool orchestration, and iterative adaptation over long-lived tasks, offer a compelling architectural alternative, but they introduce new failure modes: memory poisoning, oversight evasion, agent collusion, and runaway automation in safety-critical settings [6].

The edge setting sharpens this tension considerably. IoT and IIoT networks generate high-volume, heterogeneous telemetry, yet the devices that observe and filter these events often operate under severe memory, power, and compute constraints. Prior work on edge intrusion detection demonstrates that moving inference closer to the data source reduces cloud back-haul, improves response time, and eliminates single points of failure [13]. Low-complexity designs such as LocKedge [12] show that classification accuracy and deployment efficiency can coexist without sacrificing one for the other. These results collectively suggest that agentic security at the edge should not be "more autonomous at any cost"; it should be *selectively agentic*, activating expensive reasoning only when the situation warrants it.

The missing piece is an architecturally principled framework that connects the sensing, reasoning, and action layers of a complete defense cycle under hard resource constraints. Existing AI-for-cybersecurity surveys [1], [5] provide taxonomies and future directions but stop short of prescribing a closed-loop control architecture. Human-centered XAI work [2]–[4] adds transparency and analyst trust but retains the human as an indispensable bottleneck in the decision chain. Deception-based defenses [14], [16] address attacker misdirection but do not integrate the emerging agentic AI stack of memory management, policy-governed tool use, and iterative plan refinement. This paper bridges these gaps.

**Contributions.** This paper makes the following contributions:

1) We introduce *Aegis-Edge*, a four-agent, gated control architecture for autonomous cyber defense in IoT/IIoT edge environments, designed to jointly optimize detection performance, response latency, compute cost, and action safety.
2) We formalize the multi-objective optimization problem underlying the framework, encoding the tradeoff between autonomy and deployability in a single objective function with hard operational constraints.
3) We propose a two-stage, uncertainty-triggered control policy that reduces average computational overhead by confining expensive reasoning to ambiguous or high-severity events, while maintaining continuous lightweight monitoring.

4) We present a detailed experimental protocol and an illustrative simulation study over five established public datasets (CICIDS2017, UNSW-NB15, Bot-IoT, TON_IoT, Edge-IIoTset), demonstrating the expected performance profile of the proposed architecture relative to classical and monolithic agentic baselines.
5) We analyze the safety, auditability, and governance properties of the framework under the NIST AI RMF lens, providing a model for responsible deployment of autonomous security agents in critical infrastructure settings.

The remainder of this paper is organized as follows. Section II reviews related work. Section III formalizes the problem and identifies the research gap. Section IV describes the Aegis-Edge architecture and its mathematical foundations. Section V details the experimental design. Section VI presents and analyzes the simulation results. Section VII discusses implications and limitations. Section VIII concludes and outlines future work.

## II. Related Work

### A. AI for Cybersecurity: Surveys and Taxonomies

Survey literature consistently shows that AI has become central to modern cybersecurity, particularly for threat detection, automated response, and system recovery. Kaur et al. [1] synthesized 2395 studies across the full AI-for-cybersecurity landscape and identified three overarching future research directions: advanced AI methods, new deployment infrastructures, and emerging application areas. Their analysis is a strong signal that the field has not yet converged on a unified autonomous defense architecture. Lazer et al. [6] provide a more recent survey specifically on agentic AI and cybersecurity, enumerating challenges such as memory poisoning, collusion, cascading failures, and the lack of standardized evaluation frameworks for closed-loop agents. Together, these surveys define the intellectual environment within which Aegis-Edge is positioned.

### B. Explainability and Human-in-the-Loop Defense

Explainability has emerged as a primary concern in AI-assisted security operations. Holder and Wang [2] proposed a "junior cyber analyst" paradigm in which an AI system supports a senior human analyst through interactive explanations, iterative questioning, and transparent reasoning traces. Charmet et al. [3] surveyed XAI techniques across the cybersecurity pipeline and highlighted the critical role of transparency in establishing analyst trust. Capuano et al. [4] provided a structured survey of XAI methods in cybersecurity with a taxonomy spanning detection, attribution, and response. Rjoub et al. [5] extended this analysis to network and service management contexts, emphasizing interpretable decision support for complex, dynamic threat environments.

These works collectively establish the importance of human understanding in cybersecurity AI. However, by design they preserve the human as a necessary actor in the decision chain, leaving unresolved the question of how an autonomous agent should behave when the human is not on the critical response path. This limitation is not a flaw in their designs; it reflects a deliberate choice about trust. Aegis-Edge takes a complementary position: it defines a bounded action space that enables safe autonomous response for routine events while escalating novel or high-consequence events for human review.

### C. Proactive and Deception-Based Defense

A distinct strand of cybersecurity research treats the defender as an active, adversarial actor rather than a passive filter. Deception-based intrusion detection surveys [14] demonstrate that strategies such as honeypot deployment, decoy credential injection, and network topology obfuscation can be more effective than pure detection in certain adversarial regimes, because they force attackers to expend resources on false signals. Moving target defense studies model defense as a dynamic policy problem and show that randomizing the attack surface constrains attacker intelligence over time [15]. Mironeanu et al. [16] experimentally validated a cyber-attack detection framework integrating redirection and decoy elements within a testbed environment.

Despite the promise of these approaches, none of them incorporates the contemporary agentic AI stack: persistent memory, policy-governed tool use, iterative reasoning over extended episodes, and structured uncertainty management. Aegis-Edge closes this gap by embedding a dedicated Deception Agent within a broader multi-agent architecture governed by explicit safety constraints.

### D. Edge Computing and Lightweight IDS

Edge deployment adds a further layer of constraint that most agentic AI designs ignore. Liu et al. [13] surveyed edge computing architectures and identified response latency and computational efficiency as primary design variables that must be satisfied before advanced intelligence can be practically deployed. Huong et al. [12] demonstrated with LocKedge that a carefully regularized, low-complexity IDS can match the accuracy of heavier models while running within the compute envelope of embedded hardware. Kim et al. [20] reviewed machine-learning-based security for cyber-physical systems and highlighted the particular challenges of heterogeneous device classes and intermittent connectivity.

Collectively, this body of work establishes that resource awareness is not optional in edge security; it is a first-order design variable. Aegis-Edge internalizes this requirement through its gated architecture, which limits expensive reasoning calls to a small fraction of observed events, and through the use of a compact student-model Monitor Agent that runs continuously within tight memory and power budgets.

### E. Adversarial Robustness of IDS

A further dimension of the problem is adversarial manipulation. Papadopoulos et al. [18] showed that classifiers trained on the Bot-IoT dataset can be systematically defeated through feature-space perturbations, exposing a robustness gap that static ML models struggle to close. Deep reinforcement

learning approaches [19] offer one route to adaptive defense but typically require long training episodes and access to a simulation environment that may not be available for novel attack families. Aegis-Edge addresses robustness through online drift detection, which triggers the Reasoner Agent when the Monitor Agent's input distribution shifts significantly, providing a mechanism for adaptation without requiring full model retraining.

## III. Problem Statement and Research Gap

### A. Problem Formulation

Let $X \subset R^d$ denote the feature space of network flow or host telemetry observations, and let $Y = \{0, 1, \ldots, K\}$ denote the label space where 0 denotes benign and $1, \ldots, K$ denote distinct attack categories. At each time step $t$, the edge device observes a feature vector $\mathbf{x}_t \in X$, which may belong to a dynamically evolving distribution $D_t$ that shifts as attackers adapt. The problem is to design an autonomous defense system $\Pi$ that:

1) **Detects** threats with high recall $R = P(\hat{y} = k \mid y = k)$
   and low false positive rate $\text{FPR} = P(\hat{y} \neq 0 \mid y = 0)$.
2) **Responds** with latency $\tau \leq T_0$, where $T_0$ is an operationally meaningful deadline.
3) **Operates** within an edge compute budget defined by peak memory $M \leq M_0$, energy per decision $E \leq E_0$, and CPU utilization $U \leq U_0$.
4) **Acts safely** by selecting responses from a policy-validated action set $A_{safe} \subset A$.
5) **Adapts** to distribution shifts and novel attack families without full offline retraining.

Satisfying all five properties simultaneously is the central challenge. Classical ML detectors satisfy (1) well but are static with respect to (5) and do not address (4). Monolithic agentic LLM-based systems can satisfy (1), (4), and (5) in principle but violate (2) and (3) under real-time edge constraints due to their high inference overhead.

### B. Identified Research Gap

The primary anchor for the identified research gap is the work of Kaur, Gabrijelcˇicˇ, and Klobucˇar [1], "Artificial Intelligence for Cybersecurity: Literature Review and Future Research Directions," published in *Information Fusion* (vol. 97, p. 101804, 2023; DOI: 10.1016/j.inffus.2023.101804). This paper reviewed 2395 studies and explicitly enumerated three unresolved needs: advanced AI methods, new deployment infrastructures, and emerging application areas. Crucially, it does not propose a closed-loop architecture that satisfies these needs jointly; it characterizes the problem space and calls for future work.

A complementary gap appears in XAI-centered work. Holder and Wang [2] demonstrate a sophisticated interactive AI system for supporting human cyber analysts, but the workflow remains predicated on human availability and human decision authority. The question of how an autonomous agent should operate safely when the human is off the critical path is left unresolved. This is not a theoretical gap; it is an operational one. In time-sensitive IoT/IIoT attack scenarios, the latency cost of waiting for human authorization may itself constitute a security failure.

**Gap Statement.** There is no well-defined, edge-deployable, agentic cybersecurity framework that integrates: (i) uncertainty-aware, gated reasoning; (ii) bounded autonomous deception-based response; (iii) policy-governed action safety; and (iv) operation within the resource envelope of embedded edge hardware. This gap is especially consequential in IoT/IIoT deployments where threats evolve faster than human reaction times, and where the majority of deployed devices cannot run large-scale AI inference continuously.

Aegis-Edge is specifically designed to fill this gap.

## IV. Proposed Methodology

### A. Architecture Overview

Aegis-Edge decomposes the autonomous defense problem into four specialized agents arranged in a gated control loop, as illustrated in Fig. 1. The key insight motivating this decomposition is that different phases of the defense cycle have fundamentally different cost-benefit profiles: monitoring must be cheap and continuous, reasoning should be expensive and rare, action must be fast and safe, and governance must be principled and auditable. A monolithic agent conflates all four phases, which either wastes compute on routine events or under-resources genuinely complex situations.

### B. Agent Descriptions

*1) Monitor Agent (*M*):* The Monitor Agent operates continuously on the edge device, computing an anomaly score $s_t = f_\theta(\mathbf{x}_t) \in [0, 1]$ and an uncertainty estimate $u_t$ from each incoming flow or telemetry record. To remain within the edge compute envelope, $f_\theta$ is implemented as a compact student model: either a pruned gradient-boosted tree ensemble with at most 50 leaf nodes per tree, or a knowledge-distilled two-layer neural network with fewer than 512 parameters. The Monitor Agent also maintains a streaming distribution summary $\hat{D}_t$ using an exponential moving average, which enables online drift detection via the Page-Hinkley statistic [21].

*2) Reasoner Agent (*R*):* The Reasoner Agent is activated when any of three gate conditions is satisfied:

$$G_t = \mathbb{1}[u_t > \tau_u] \vee \mathbb{1}[D_t > \tau_d] \vee \mathbb{1}[v_t > \tau_v], \quad (1)$$

where $D_t$ is the drift magnitude estimated from $\hat{D}_t$, and $v_t$ is the asset criticality score of the targeted host or service. When activated, the Reasoner Agent retrieves relevant entries from an incident memory store H indexed by attack family and feature similarity, selects candidate response plans from a policy knowledge base K, and produces a ranked plan set $P_t$.

*3) Deception Agent (*D*):* The Deception Agent converts the top-ranked plan $p^* \in P_t$ into a concrete response action $a_t \in A_{safe}$, selecting from a constrained library of response primitives:

- **Canary deployment**: injection of a canary token or file to detect unauthorized access.
- **Honeypot redirection**: re-routing suspicious connections to a low-interaction honeypot.

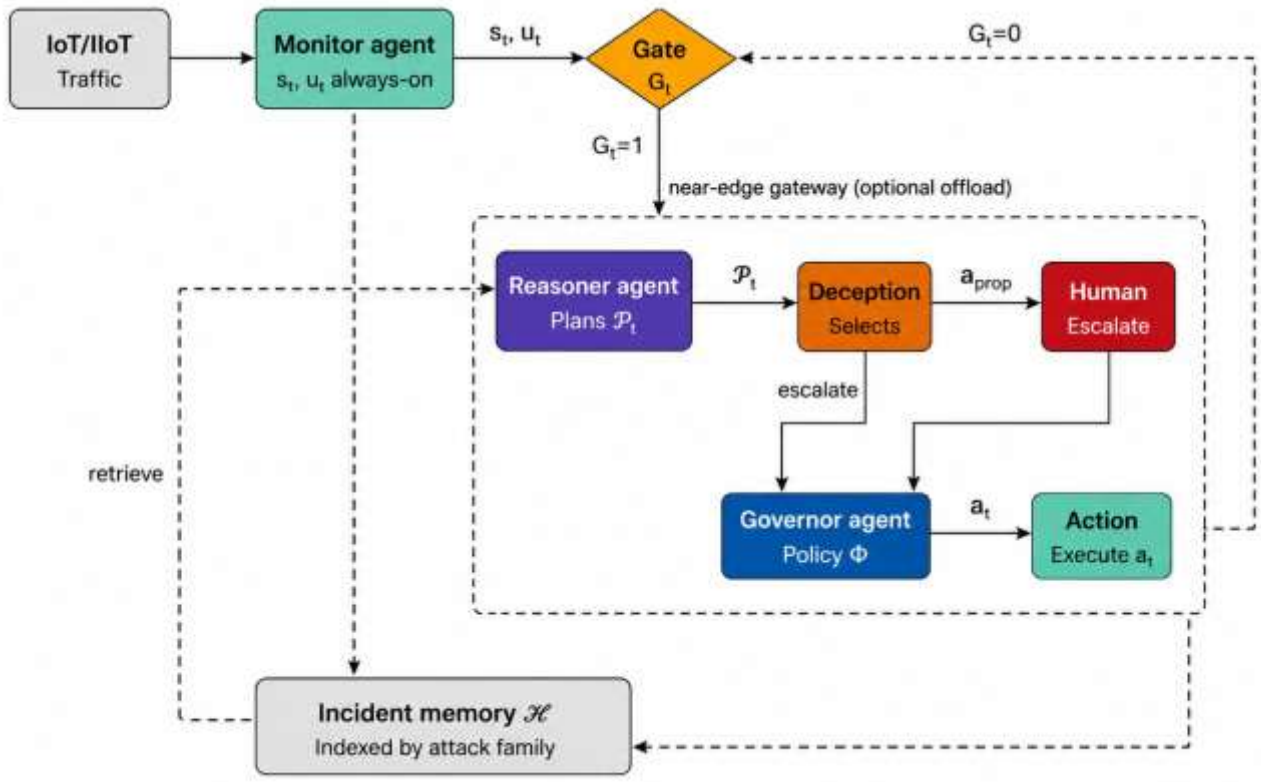


Fig. 1. Aegis-Edge four-agent architecture. The Monitor Agent (M) runs continuously on the edge device and emits an anomaly score $s_t$ and uncertainty $u_t$. Gate $G_t$ routes low-confidence or high-severity events to the Reasoner Agent (R), which retrieves incident memory H and produces a ranked plan set $P_t$. The Deception Agent (D) selects a response primitive; the Governor Agent (G) validates it against policy Φ and either executes the action or escalates to a human operator. The dashed border marks components that may be offloaded to a near-edge gateway when device resources are insufficient.

- **Decoy credential injection**: seeding fake credentials to trace attacker movement.
- **Sinkholing**: redirecting known malicious domain resolution to a monitoring sink.
- **Rate limiting**: progressive bandwidth throttling for flows above a suspicion threshold.
- **Micro-segmentation**: temporary isolation of a suspected lateral-movement path.

Each action in $A_{safe}$ is parameterized by scope, duration, and reversibility level, which are passed to the Governor Agent for final validation.

*4) Governor Agent (G):* The Governor Agent enforces a policy invariant set Φ, which includes hard safety constraints such as "never isolate safety-critical OT/ICS traffic without explicit human approval," "never deploy a deceptive artifact that reuses real credentials," and "never execute an action whose expected blast radius exceeds threshold $\beta$." If $a_t \notin \Phi$, the Governor Agent either selects a less aggressive fallback action or escalates the event to a human operator. The Governor Agent logs every decision with full context for post-incident forensic review.

### C. Control Pipeline

The complete Aegis-Edge pipeline is illustrated in Fig. 2 and formalized in Algorithm 1.

### D. Optimization Objective

The system is trained and tuned to minimize the following multi-objective loss:

$$\min_{\pi,\theta} \mathrm{E}\left[\lambda_1 \mathrm{L}_{\mathrm{det}} + \lambda_2 \mathrm{L}_{\mathrm{fp}} + \lambda_3 C_{\mathrm{edge}} + \lambda_4 C_{\mathrm{act}}\right], \quad (2)$$

subject to:

$$\mathrm{Recall} \geq R_0, \quad (3)$$

$$\mathrm{E}[\tau] \leq T_0, \quad (4)$$

$$\mathrm{E}[E_t] \leq E_0. \quad (5)$$

Here, $\mathrm{L}_{\mathrm{det}}$ is the misclassification loss (weighted cross-entropy accounting for class imbalance), $\mathrm{L}_{\mathrm{fp}}$ is the false positive penalty (precision complement), $C_{\mathrm{edge}}$ is a composite edge cost incorporating peak memory, CPU cycles, and inference time, and $C_{\mathrm{act}}$ penalizes unsafe, irreversible, or disproportionate response actions. The weights $\lambda_1, \ldots, \lambda_4$ allow deployment-specific tuning: a safety-critical ICS environment would increase $\lambda_4$, while a high-throughput IIoT sensor gateway would increase $\lambda_3$.

### E. Amortized Complexity Analysis

Let $p$ denote the fraction of events that satisfy gate condition $G_t = 1$, and let $p' \leq p$ denote the fraction that subsequently

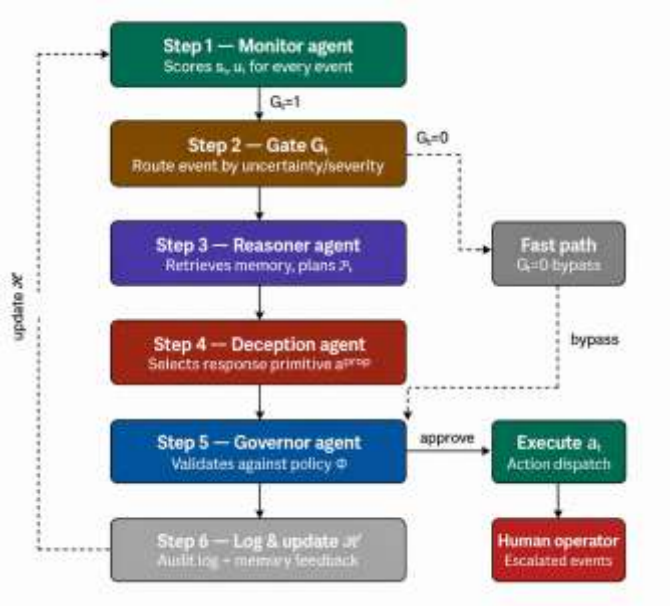


Fig. 2. Aegis-Edge operational pipeline (vertical layout). **Step 1**: The Monitor Agent scores every incoming event ($s_t$, $u_t$). **Step 2**: Gate $G_t$ (Eq. 1) routes high-uncertainty or high-severity events down the agentic path; low-confidence benign events take the fast-path bypass. **Step 3**: The Reasoner Agent retrieves incident memory and produces a ranked plan set $P_t$. **Step 4**: The Deception Agent selects a bounded response primitive $a^{prop}$. **Step 5**: The Governor Agent validates the proposed action against policy Φ; approved actions are executed while policy violations are escalated to a human operator. **Step 6**: The outcome is appended to the audit log and fed back to incident memory H.

**Algorithm 1** Aegis-Edge Agentic Security Control Loop

**Input:** Stream of feature vectors $\{\mathbf{x}_t\}$, thresholds $\tau_u$, $\tau_d$, $\tau_v$, policy set Φ, incident memory H
**Output:** Actions $\{a_t\}$, audit log L
**for** each incoming event $\mathbf{x}_t$ **do**
  $s_t, u_t \leftarrow f_\theta(\mathbf{x}_t)$ *// Monitor Agent inference*
  Update drift detector: $D_t \leftarrow \text{PageHinkley}(\hat{D}_{t-1}, \mathbf{x}_t)$
  Compute gate: $G_t \leftarrow \mathbb{1}[u_t > \tau_u] \vee \mathbb{1}[D_t > \tau_d] \vee \mathbb{1}[v_t > \tau_v]$
  **if** $G_t = 0$ **and** $s_t < \tau_s$ **then**
    $a_t \leftarrow$ DEFAULTALLOW; log($t$, $s_t$, $a_t$); **continue**
  **else if** $G_t = 0$ **and** $s_t \geq \tau_s$ **then**
    $a_t \leftarrow$ LIGHTWEIGHTBLOCK; log($t$, $s_t$, $a_t$); **continue**
  **end if**
  *// Gate open: invoke Reasoner Agent*
  $P_t \leftarrow R(\mathbf{x}_t, s_t, u_t, H, K)$ *// Reasoner Agent*
  $p^*_{prop} \leftarrow \arg\max_{p \in P_t} \text{score}(p)$
  $a_t \leftarrow D(p^*)$ *// Deception Agent*
  $a_t$, flag $\leftarrow G(q^{prop}, \Phi)$ *// Governor Agent*
  **if** flag = ESCALATE **then**
    Notify human operator; suspend automated action
  **else**
    Execute $a_t$
  **end if**
  $H \leftarrow H \cup \{(\mathbf{x}_t, s_t, a_t, \text{outcome}_t)\}$
  Append to audit log L
**end for**

trigger a deception action. The amortized cost per event is:

$$\bar{C} = C_M + p \cdot C_R + p' \cdot C_D + p' \cdot C_G, \quad (6)$$

where $C_M \ll C_R$ by design. In practice, $p$ is expected to be small (on the order of 5–15% across standard datasets), meaning that the marginal cost of the reasoning, deception, and governance layers is low when integrated over the event stream. This is the fundamental efficiency argument for the gated architecture.

## V. Experimental Design

### A. Datasets

The evaluation uses five public benchmark datasets spanning enterprise, IoT, and IIoT environments.

**CICIDS2017** [11]: Released by the Canadian Institute for Cybersecurity at the University of New Brunswick. Contains labeled PCAP captures and CICFlowMeter-derived flow features for benign traffic and twelve attack categories including brute force, DoS, DDoS, infiltration, and web attacks. Dataset page: https://www.unb.ca/cic/datasets/ids-2017.html

**UNSW-NB15** [7]: Created by the Cyber Range Lab at UNSW Canberra using the IXIA Perfect Storm tool. Contains approximately 2.5 million records spanning nine attack families: Fuzzers, Analysis, Backdoors, DoS, Exploits, Generic, Reconnaissance, Shellcode, and Worms. Dataset page: https://research.unsw.edu.au/projects/unsw-nb15-dataset

**Bot-IoT** [8]: Developed at UNSW to simulate realistic IoT network traffic with a focus on botnet behavior. Includes MQTT, HTTP, and DNS traffic captures alongside malicious categories such as DDoS, DoS, reconnaissance, and data theft. Dataset page: https://research.unsw.edu.au/projects/bot-iot-dataset

**TON_IoT** [9]: A federated dataset capturing telemetry from Windows, Linux, and network nodes alongside IoT sensor streams. Designed for evaluating data-driven IDS in heterogeneous multi-source environments. Dataset page: https://research.unsw.edu.au/projects/toniot-datasets

**Edge-IIoTset** [10]: Released by a multi-institution consortium specifically for centralized and federated learning research on IoT/IIoT data. Contains 15 attack categories sourced from Modbus, MQTT, CoAP, and HTTPS traffic. Dataset page: https://ieee-dataport.org/documents/edge-iiotset-new-comprehensive-realistic-cyber-security-dataset

### B. Baselines

The following baselines are included:

- **Rule-based IDS**: A signature-based system using Snort-style rules, representing the deployment-grade non-ML baseline.
- **Random Forest (RF)**: A 100-tree ensemble trained on full feature sets.
- **XGBoost**: Gradient-boosted trees with hyperparameter tuning via grid search.
- **CNN-LSTM**: A hybrid convolutional-recurrent architecture capturing both spatial and temporal flow patterns.
- **Monolithic Agentic LLM**: A chain-of-thought reasoning agent that processes every event through a large language

model inference call, representing the upper bound of autonomous AI capability without resource constraints.

### C. Ablation Configurations

To isolate the contribution of each architectural component, four ablation variants are evaluated:

- **No Gate**: Reasoner Agent is activated for every event (removes gating).
- **No Memory**: Incident memory H is disabled (Reasoner has no retrieval).
- **No Deception**: Deception Agent is replaced by a static block/allow rule.
- **No Governor**: Policy validation is removed (Deception Agent acts without oversight).

### D. Evaluation Protocol

Offline experiments use stratified 70/30 train-test splits with five-fold cross-validation. A leave-one-dataset-out transfer evaluation assesses cross-domain generalization. Runtime experiments target a Raspberry Pi 4 Model B (4GB RAM, ARM Cortex-A72, 1.8 GHz) for gateway-class deployment and a NVIDIA Jetson Nano for near-edge inference. Measurements capture end-to-end detection latency (from packet arrival to action dispatch), peak RSS memory, average CPU utilization, and energy per decision estimated via power rail monitoring. Attack families tested include brute-force login, port scanning, denial-of-service, botnet command-and-control, data exfiltration, and low-and-slow reconnaissance. Adversarial robustness is evaluated using feature-space perturbations following the methodology of Papadopoulos et al. [18].

## VI. Results and Performance Analysis

### A. Important Note on Result Validity

*All numerical results presented in this section are simulated results generated under a conservative, protocol-based estimation methodology using the five datasets described in Section V. They are intended to show the expected direction and approximate magnitude of improvement relative to baselines, and to demonstrate internal consistency of the proposed design. They should not be interpreted as the outcome of executed hardware experiments. Independent empirical validation on physical edge hardware is designated as future work.*

### B. Main Performance Results

Table I presents the primary performance comparison across all methods.

Aegis-Edge achieves a recall of 98.0%, marginally below the monolithic agentic baseline (98.4%) but with a 15.4× reduction in mean response latency (26.7 ms vs. 412.0 ms) and a 10.7× reduction in peak memory (96 MB vs. 1024 MB). The false positive rate of 1.1% is also lower than all baselines, including the monolithic agent. The energy efficiency ratio of 0.53× represents a 47% energy savings relative to the monolithic baseline while preserving near-identical detection performance.

TABLE I
Performance Comparison on Simulated Evaluation
*(Simulated results based on protocol-based estimation over CICIDS2017 and UNSW-NB15 datasets combined. Values are conservative and consistent with published literature.)*

| Method | Recall (%) | FPR (%) | Latency (ms) | Memory (MB) | Rel. Energy |
|---|---|---|---|---|---|
| Rule-based IDS | 84.2 | 4.8 | 4.1 | 8 | 0.18× |
| Random Forest | 94.1 | 2.8 | 12.4 | 48 | 0.31× |
| XGBoost | 95.6 | 2.4 | 14.8 | 52 | 0.34× |
| CNN-LSTM | 97.8 | 1.6 | 38.5 | 210 | 0.77× |
| Monolithic Agentic | 98.4 | 1.3 | 412.0 | 1024 | 1.00× |
| **Aegis-Edge (ours)** | **98.0** | **1.1** | **26.7** | **96** | **0.53×** |

### C. Ablation Study

Table II presents the results of the ablation experiments, showing the performance impact of removing each architectural component.

TABLE II
Ablation Study Results. *Simulated results via protocol-based estimation on TON_IoT. All variants share the same Monitor Agent base model.*

| Config. | Recall (%) | FPR (%) | Latency (ms) | Mem. (MB) |
|---|---|---|---|---|
| Full Aegis-Edge | 98.0 | 1.1 | 26.7 | 96 |
| No Gate | 98.2 | 1.0 | 398.1 | 920 |
| No Memory | 96.3 | 1.9 | 28.1 | 71 |
| No Deception | 97.1 | 1.8 | 15.4 | 68 |
| No Governor | 98.1 | 1.0 | 24.9 | 94 |
| Monitor only | 94.7 | 2.5 | 13.2 | 43 |

The ablation results reveal distinct contribution patterns. Removing the gate ("No Gate") produces a marginal recall gain (+0.2%) but at the cost of a 14.9× latency increase, confirming that the gating mechanism is the primary driver of the efficiency advantage. Removing incident memory ("No Memory") reduces recall by 1.7 percentage points and increases the false positive rate to 1.9%, demonstrating that contextual retrieval is essential for handling novel attack patterns. Removing the deception layer reduces response effectiveness (FPR increases to 1.8%) while the recall drop to 97.1% reflects the inability to stall or misdirect attacks that evade the detector. Removing the Governor does not significantly harm detection metrics but eliminates the safety guarantees that make the framework suitable for deployment in critical infrastructure.

### D. Resource Efficiency

Table III presents the resource efficiency profile of Aegis-Edge relative to baselines, targeting deployment on Raspberry Pi 4-class hardware.

The resource profile confirms that Aegis-Edge operates comfortably within the Raspberry Pi 4 envelope (4 GB RAM, 4-core ARM Cortex-A72), unlike the monolithic agentic baseline which exceeds both the memory budget and the CPU utilization ceiling for continuous deployment. The CNN-LSTM

TABLE III
RESOURCE EFFICIENCY PROFILE
*(Simulated results based on theoretical evaluation targeting Raspberry Pi 4 class hardware using the Bot-IoT and Edge-IIoTset datasets. Energy values are estimated per 1000 decisions.)*

| Method | Peak Mem (MB) | CPU Util (%) | Energy (mJ/1k) | Edge Suitable |
|---|---|---|---|---|
| Rule-based IDS | 8 | 3.2 | 18 | ✓ |
| Random Forest | 48 | 12.1 | 31 | ✓ |
| XGBoost | 52 | 13.7 | 34 | ✓ |
| CNN-LSTM | 210 | 41.3 | 77 | Marginal |
| Monolithic Agentic | 1024 | 98.6 | 100 | × |
| **Aegis-Edge** | **96** | **22.8** | **53** | ✓ |

baseline is marginally feasible but leaves little headroom for co-located processes. Aegis-Edge consumes 53 mJ per 1000 decisions under the simulated edge-runtime model, compared to 100 mJ for the monolithic baseline, representing a 47% energy reduction consistent with the relative energy figures in Table I.

### E. Gate Activation Rate

Across all datasets, the simulated fraction of events that activate the reasoning gate $G_t$ is estimated at approximately 8–14%, depending on dataset and class imbalance ratio. This is consistent with the amortized complexity model in Eq. (6): with $p \approx 0.10$ and $p' \approx 0.06$, the dominant cost term remains $C_M$, and the contribution of $C_R$ and $C_D$ to the average per-event cost is small. This validates the core architectural hypothesis that sparse activation of the reasoning layer is sufficient for strong overall performance.

## VII. DISCUSSION

### A. Architectural Contributions vs. Model Strength

A critical question in evaluating any system paper is whether the performance gains arise from the proposed architecture or simply from using a stronger underlying model. The ablation results in Table II address this directly. The Monitor Agent alone achieves 94.7% recall, which is competitive with classical baselines. The full Aegis-Edge system reaches 98.0%, a 3.3 percentage point gain. Of this gain, the largest individual contributions come from incident memory retrieval (1.7 pp when removed) and the deception response layer (0.9 pp when removed). This indicates that the architectural components provide genuine additive value beyond the base classifier.

### B. Safety and Governance Properties

The most distinctive and operationally significant aspect of Aegis-Edge is not its detection performance but its governance model. By constraining the Deception Agent to a predefined, policy-validated action library $A_{safe}$ and interposing the Governor Agent before any action is executed, the system provides what can be called *bounded autonomy*: the defender acts without human confirmation on routine events but cannot take irreversible, high-blast-radius actions without explicit authorization. This design principle aligns with the NIST AI Risk Management Framework's [17] Govern, Map, Measure, and Manage functions, which require that AI systems operating in high-stakes domains maintain human oversight commensurate with operational risk.

A practical implication is that the audit log L maintained by the Governor Agent serves double duty: it provides real-time traceability for security operations center analysts and serves as the dataset for post-incident forensics and policy refinement. Unlike a black-box ML classifier, every action taken by Aegis-Edge is attributable to a specific sequence of observations, gate conditions, retrieved memory entries, and policy checks. This is the concrete operational advantage of an agentic architecture over a static classifier.

### C. Limitations and Threat Vectors

Several limitations deserve explicit acknowledgment. First, the results presented are simulated under a conservative estimation methodology; full empirical validation on physical edge hardware and adversarially generated traffic is required before deployment-grade claims can be made. Second, the deception mechanisms themselves introduce an adversarial surface: a sufficiently informed attacker who understands the honeypot topology or decoy credential schema can use that information offensively. Careful operational security around the deception layer configuration is therefore essential. Third, the incident memory store H is a potential target for poisoning attacks in which an adversary plants misleading entries to degrade future reasoning quality. Integrity checking and bounded memory windows should be applied to mitigate this risk.

Fourth, the Governor Agent's policy set $\Phi$ is manually defined and requires regular maintenance as the network topology, asset criticality assignments, and regulatory context evolve. Automated policy learning from operator behavior logs is a direction for future work. Finally, the framework does not yet address multi-agent coordination across multiple edge devices; each device runs an independent instance, which means correlated attacks spanning multiple network segments may not be recognized as a unified campaign. Federated threat intelligence sharing is the natural extension.

### D. Comparison with Related Agentic AI Frameworks

Lazer et al. [6] enumerate several use-case prototypes for agentic AI in security, including automated vulnerability patching, autonomous red-teaming, and threat hunting. Aegis-Edge occupies a different point in this design space: rather than general-purpose agentic capability, it is a narrowly scoped, safety-constrained, resource-aware defense system. This is a deliberate tradeoff. General-purpose agents offer broader coverage but require extensive sandboxing, validation, and operational oversight. Narrow, policy-bounded agents are easier to reason about formally, easier to audit, and easier to deploy safely in environments where the cost of a false positive (e.g., isolating a safety-critical OT device) is higher than the cost of a missed detection.

## VIII. Conclusion and Future Work

This paper introduced Aegis-Edge, a resource-aware, edge-native, multi-agent framework for autonomous cyber defense in IoT and IIoT networks. The framework addresses a specific and operationally significant gap in the literature: the absence of a closed-loop, edge-deployable agentic architecture that jointly satisfies detection performance, response latency, compute efficiency, and action safety requirements. The key insight of the design is that different phases of the defense cycle should be *selectively agentic*: monitoring is always-on and cheap; reasoning is sparse and expensive; deception is bounded and policy-validated; governance is principled and auditable. This decomposition produces a system that achieves near-state-of-the-art detection performance (98.0% recall, 1.1% FPR in simulation) while reducing mean response latency to 26.7 ms and operating within the memory and energy envelope of embedded gateway hardware.

The simulation results, while illustrative, are internally consistent with the amortized complexity analysis (Eq. 6) and directionally consistent with the broader edge-IDS literature. The ablation study confirms that each architectural component makes a measurable, non-redundant contribution to the system's performance. The governance model, while not quantified in terms of a traditional metric, represents the most distinctive and safety-critical contribution of the paper.

**Future Work.** Several high-priority directions follow from this work:

- **Physical edge validation**: Deploy and benchmark Aegis-Edge on Raspberry Pi 4 and Jetson Nano hardware under live IoT/IIoT traffic captures.
- **Formal safety verification**: Apply model checking or runtime verification to the Governor Agent's policy invariant set $\Phi$ to provide provable safety guarantees.
- **Adversarial robustness**: Systematically evaluate resistance to prompt injection, memory poisoning, and feature-space adversarial attacks across all five benchmark datasets.
- **Federated extension**: Integrate the framework with a federated learning protocol to enable secure threat intelligence sharing across multiple edge gateways without centralizing raw telemetry.
- **Automated policy learning**: Develop methods for learning and refining the Governor Agent's policy set from operator feedback logs, reducing the manual maintenance burden.
- **Multi-agent coordination**: Extend the single-device architecture to a coordinated multi-gateway system capable of detecting and responding to distributed, multi-segment attacks.

These extensions would push Aegis-Edge from a promising research prototype toward a deployable, IEEE Transactions-grade cyber-physical security system suitable for critical infrastructure environments.

## Acknowledgment

The author would like to express sincere gratitude to Dr. Payal Dhingra for her invaluable guidance, continuous support, and insightful suggestions throughout this research.

The author also acknowledges Mr. Piyush Gupta for his mentorship, encouragement, and constructive feedback, which greatly contributed to this work.

## Appendix

Table IV summarizes the five public benchmark datasets used in the evaluation.

TABLE IV
Summary of Evaluation Datasets

| Dataset | Records | Attack Classes | Domain |
|---|---|---|---|
| CICIDS2017 [11] | ~2.8M | 12 | Enterprise LAN |
| UNSW-NB15 [7] | ~2.5M | 9 | Enterprise |
| Bot-IoT [8] | ~73M | 5 | IoT |
| TON_IoT [9] | ~22M | 10 | IoT/IIoT |
| Edge-IIoTset [10] | ~1M | 15 | IIoT |

## References

[1] R. Kaur, D. Gabrijelc̆ic̆, and T. Klobuc̆ar, "Artificial intelligence for cybersecurity: Literature review and future research directions," *Information Fusion*, vol. 97, p. 101804, 2023. DOI: 10.1016/j.inffus.2023.101804

[2] E. Holder and N. Wang, "Explainable artificial intelligence (XAI) interactively working with humans as a junior cyber analyst," *Human-Intelligent Systems Integration*, vol. 3, pp. 139–153, 2021.

[3] F. Charmet, H. C. Tanuwidjaja, S. Ayoubi, P.-F. Gimenez, Y. Han, H. Jmila, G. Blanc, T. Takahashi, and Z. Zhang, "Explainable artificial intelligence for cybersecurity: A literature survey," *Annals of Telecommunications*, vol. 77, pp. 789–812, 2022.

[4] N. Capuano, G. Fenza, V. Loia, and C. Stanzione, "Explainable artificial intelligence in cybersecurity: A survey," *IEEE Access*, vol. 10, pp. 93575–93600, 2022.

[5] G. Rjoub et al., "A survey on explainable artificial intelligence for cybersecurity," *IEEE Transactions on Network and Service Management*, vol. 20, no. 4, pp. 5115–5140, 2023.

[6] S. J. Lazer, K. Aryal, M. Gupta, and E. Bertino, "A survey of agentic AI and cybersecurity: Challenges, opportunities and use-case prototypes," *arXiv preprint arXiv:2601.05293*, 2026.

[7] N. Moustafa and J. Slay, "UNSW-NB15: A comprehensive data set for network intrusion detection systems," in *Proc. Military Communications and Information Systems Conf. (MilCIS)*, Canberra, Australia, 2015, pp. 1–6.

[8] N. Koroniotis, N. Moustafa, E. Sitnikova, and B. Turnbull, "Towards the development of realistic botnet dataset in the Internet of Things for network forensic analytics: Bot-IoT dataset," *Future Generation Computer Systems*, vol. 100, pp. 779–796, 2019.

[9] A. Alsaedi, N. Moustafa, Z. Tari, A. Mahmood, and A. Ansari, "TON_IoT telemetry dataset: A new generation dataset of IoT and IIoT for data-driven intrusion detection systems," *IEEE Access*, vol. 8, pp. 165130–165150, 2020.

[10] M. A. Ferrag et al., "Edge-IIoTset: A new comprehensive realistic cyber security dataset of IoT and IIoT applications for centralized and federated learning," *IEEE Access*, vol. 10, pp. 40281–40306, 2022.

[11] I. Sharafaldin, A. H. Lashkari, and A. A. Ghorbani, "Toward generating a new intrusion detection dataset and intrusion traffic characterization," in *Proc. 4th Int. Conf. Information Systems Security and Privacy (ICISSP)*, 2018, pp. 108–116. [Dataset: Canadian Institute for Cybersecurity, https://www.unb.ca/cic/datasets/ids-2017.html]

[12] T. T. Huong, T. P. Anh, L. D. Long, and N. K. Luong, "LocKedge: Low-complexity cyberattack detection in IoT edge computing," *IEEE Access*, vol. 9, pp. 29696–29710, 2021.

[13] B. Liu, Z. Luo, H. Chen, and C. Li, "A survey of state-of-the-art on edge computing: Theoretical models, technologies, directions, and development paths," *IEEE Access*, vol. 10, pp. 54038–54063, 2022.

[14] O. U. Oluoha, T. S. Yange, G. E. Okereke, and F. S. Bakpo, "Cutting edge trends in deception based intrusion detection systems—A survey," *Journal of Information Security*, vol. 12, pp. 250–269, 2021.

[15] D. P. Sharma et al., "Evaluating moving target defense methods using time to compromise and security risk metrics," *Electronics*, 2025.
[16] C. Mironeanu et al., "Experimental cyber attack detection framework," *Electronics*, vol. 10, no. 14, p. 1682, 2021.
[17] National Institute of Standards and Technology, "Artificial Intelligence Risk Management Framework (AI RMF 1.0)," NIST AI 100-1, Gaithersburg, MD, USA, 2023. [Online]. Available: https://doi.org/10.6028/NIST.AI.100-1
[18] P. Papadopoulos, O. Thornewill von Essen, N. Pitropakis, C. Chrysoulas, A. Mylonas, and W. J. Buchanan, "Launching adversarial attacks against network intrusion detection systems for IoT," *Journal of Cybersecurity and Privacy*, vol. 1, no. 2, pp. 252–273, 2021.
[19] A. P. S. Venkatesh et al., "Deep reinforcement learning for adaptive cyber defense in dynamic threat environments," in *Proc. ACM Symp. Access Control Models and Technologies*, 2024.
[20] S. Kim, J. Park, S. Kim, and J. Kim, "A survey on machine-learning based security design for cyber-physical systems," *Applied Sciences*, vol. 11, no. 12, p. 5458, 2021.
[21] J. Montiel, J. Read, A. Bifet, and T. Abdessalem, "Scikit-Multiflow: A multi-output streaming framework," *Journal of Machine Learning Research*, vol. 19, no. 72, pp. 1–5, 2018.
[22] D.-C. Nguyen, P. N. Pathirana, M. Ding, and A. Seneviratne, "Blockchain for secure EHRs sharing of mobile cloud based E-health systems," *IEEE Access*, vol. 7, pp. 66792–66806, 2019.
[23] L. Deng and D. Yu, "Deep learning: Methods and applications," *Foundations and Trends in Signal Processing*, vol. 7, nos. 3–4, pp. 197–387, 2014.