ISSN: 1813-4890

Autonomous Threat Detection and Response through Actor–Critic Reinforcement Learning

Shaochen Ren [1], Shiyang Chen [2], Qun Zhang [3]

[1]Tandon School of Engineering, New York University, New York, NY 10012, USA

[2]College of Engineering, Texas A&M University, College Station, TX 77840, USA

[3]Department of Statistics and Biostatistics, California State University, East Bay, Hayward, CA 94542, USA

Corresponding author: Qun Zhang. qzhang46@horizon.csueastbay.edu

Abstract

The escalating sophistication of cyber threats necessitates intelligent autonomous defense mechanisms capable of real-time detection and response. Reinforcement learning (RL) has emerged as a promising paradigm for developing adaptive security systems that learn optimal defense strategies through environmental interaction. Among various RL architectures, actor-critic (AC) methods have demonstrated superior performance in continuous and complex action spaces typical of cybersecurity scenarios. This review paper provides a comprehensive analysis of actor-critic reinforcement learning applications in autonomous threat detection and response systems. We examine the theoretical foundations of AC algorithms, including advantage actor-critic (A2C), asynchronous advantage actor-critic (A3C), soft actor-critic (SAC), and deep deterministic policy gradient (DDPG) methods. The paper explores how these algorithms address critical challenges in cybersecurity, including high-dimensional state spaces, concept drift in attack patterns, delayed rewards, and the need for real-time decision-making. We analyze recent advances in network intrusion detection systems (NIDS), malware analysis, advanced persistent threat (APT) detection, and automated incident response using AC frameworks. Furthermore, we discuss integration strategies with existing security infrastructure, scalability considerations for enterprise environments, and approaches to handling adversarial attacks against learning systems. The review identifies current limitations including sample efficiency, interpretability concerns, and the reality gap between simulated training environments and production systems. We conclude by outlining promising research directions, including meta-learning approaches, multi-agent coordination, explainable reinforcement learning for security, and hybrid architectures combining AC methods with symbolic reasoning systems.

Keywords

Actor-Critic Reinforcement Learning; Autonomous Threat Detection; Cybersecurity; Network Intrusion Detection.

1. Introduction

The contemporary cybersecurity landscape confronts organizations with unprecedented challenges characterized by increasingly sophisticated attack vectors, rapidly evolving threat actors, and massive volumes of security-relevant data requiring analysis. Traditional signature-based detection systems and rule-driven response mechanisms struggle to adapt to novel attack patterns and zero-day exploits [1]. The average time to detect a breach in enterprise

networks exceeds 200 days, during which adversaries maintain persistent access and exfiltrate sensitive information [2]. This detection latency, combined with the shortage of skilled security analysts and the overwhelming number of alerts generated by conventional security tools, creates an urgent need for intelligent automation in threat detection and response operations.

Reinforcement learning (RL) represents a fundamentally different approach to cybersecurity automation by enabling systems to learn optimal defense strategies through trial-and-error interaction with the environment rather than relying on predefined rules or labeled training datasets [3]. Unlike supervised learning methods that require extensive labeled attack datasets, RL agents develop adaptive behaviors by receiving rewards for successful threat mitigation and penalties for security breaches or false positives. This learning paradigm aligns naturally with the adversarial nature of cybersecurity, where defenders must continuously adapt to evolving attacker tactics, techniques, and procedures [4].

Among various RL architectures, actor-critic (AC) methods have gained prominence due to their ability to handle continuous action spaces and provide stable learning in complex environments [5]. The AC framework combines policy-based and value-based approaches by maintaining two neural networks: an actor network that selects actions and a critic network that evaluates the quality of those actions. This architecture offers several advantages for cybersecurity applications, including reduced variance in policy gradient estimates, improved sample efficiency compared to pure policy gradient methods, and the capability to learn both deterministic and stochastic policies [6]. The actor learns to optimize the policy for selecting defensive actions such as network traffic filtering, system isolation, or threat hunting priorities, while the critic provides feedback on the expected long-term security outcomes of those decisions [7].

The application of AC reinforcement learning to autonomous threat detection and response addresses several critical requirements in modern security operations centers. First, AC methods can process high-dimensional state representations encompassing network traffic features, system logs, threat intelligence feeds, and contextual information about assets and vulnerabilities [8]. Second, they support continuous action spaces necessary for fine-grained defensive responses such as adaptive firewall rule adjustments or dynamic resource allocation for security monitoring [5]. Third, AC algorithms demonstrate robust performance in partially observable environments where complete information about attacker activities may be unavailable due to evasion techniques or limited visibility into encrypted communications [9]. Fourth, these methods can incorporate domain knowledge through reward shaping and architectural inductive biases while retaining the ability to discover novel defensive strategies not anticipated by human experts [10].

Despite promising theoretical properties and encouraging results in simulated environments, deploying AC-based autonomous defense systems in production cybersecurity infrastructure presents substantial challenges. The reality gap between training simulations and actual network environments can lead to unpredictable behaviors when deployed [11]. Security-critical applications require high reliability and interpretability, characteristics that deep RL systems often lack [12]. Adversarial machine learning attacks can potentially manipulate the learning process or exploit vulnerabilities in trained policies [13]. Sample efficiency remains a concern, as gathering sufficient real-world cybersecurity experience for training while maintaining security during the learning phase poses practical difficulties [14]. Integration with existing security orchestration, automation, and response platforms requires careful architectural design and validation [15].

This review paper provides a comprehensive examination of AC reinforcement learning methods for autonomous threat detection and response systems. We systematically analyze the theoretical foundations of AC algorithms, their specific adaptations for cybersecurity applications, empirical results from recent research, and practical considerations for

ISSN: 1813-4890

deployment. The paper synthesizes current knowledge to identify gaps, highlight successful approaches, and outline promising directions for future research at the intersection of RL and cybersecurity.

2. Literature Review

The intersection of RL and cybersecurity has evolved significantly over the past five years, with AC methods emerging as a dominant paradigm for autonomous defense systems. Early applications of RL to network security focused primarily on intrusion detection using Q-learning and deep Q-networks, but these approaches faced limitations in handling continuous action spaces and high-dimensional state representations characteristic of modern security operations [3]. The shift toward AC architectures began as researchers recognized the need for more sophisticated policy optimization techniques capable of learning nuanced defensive strategies in complex adversarial environments.

Recent literature demonstrates that AC methods provide superior performance in NIDS compared to traditional machine learning classifiers and earlier RL approaches. Research has shown that A2C algorithms can achieve detection rates exceeding 95% on benchmark datasets such as NSL-KDD and CICIDS2017 while maintaining false positive rates below 2% [16]. These results represent significant improvements over conventional anomaly detection systems that struggle to balance sensitivity and specificity in high-throughput network environments. The AC framework enables dynamic adjustment of detection thresholds based on current threat levels and organizational risk tolerance, adapting to changing operational contexts in real-time [17].

APT detection represents another domain where AC reinforcement learning has demonstrated considerable promise. APT campaigns involve multi-stage attacks that unfold over extended periods, requiring security systems to identify subtle correlations across temporally dispersed indicators of compromise [18]. Traditional detection methods rely on predetermined attack patterns and struggle with the stealthy, adaptive nature of sophisticated threat actors. Researchers have developed AC-based systems that learn to recognize APT behavior patterns through sequential decision-making processes that account for long-term dependencies in attacker actions [19]. These systems employ recurrent neural network architectures in both actor and critic networks to maintain memory of historical observations, enabling detection of coordinated attack sequences that span days or weeks [20].

The application of SAC algorithms to malware analysis and classification has garnered substantial research attention due to the continuous evolution of malicious code and the inadequacy of signature-based detection methods. SAC's entropy regularization mechanism encourages exploration of diverse defensive strategies, preventing premature convergence to suboptimal policies that might be exploited by adaptive malware [6]. Studies have demonstrated that SAC-based malware detectors can identify polymorphic and metamorphic malware variants by learning invariant behavioral patterns rather than relying on static code signatures [21]. The stochastic policies learned by SAC provide robustness against adversarial examples designed to evade detection, as the randomness in action selection makes it difficult for attackers to craft reliably evasive malware [22].

Automated incident response systems leveraging AC reinforcement learning address the critical challenge of reducing response times and minimizing human intervention in security operations. Research has explored AC methods for selecting optimal remediation actions from complex response playbooks, considering factors such as attack severity, asset criticality, business impact, and available defensive resources [23]. These systems learn to orchestrate multi-step response procedures including network isolation, forensic data collection, threat intelligence enrichment, and stakeholder notification [24]. Experimental results indicate that

ISSN: 1813-4890

AC-based response automation can reduce mean time to respond by over 60% compared to manual processes while maintaining higher consistency in following security policies [25].

The integration of DDPG algorithms with network traffic analysis has enabled development of adaptive firewall and intrusion prevention systems that dynamically optimize security policies. DDPG's ability to learn deterministic policies in continuous action spaces allows for precise control over traffic filtering rules, bandwidth allocation, and connection rate limiting [5]. Research demonstrates that DDPG-based network defense systems can learn to distinguish between legitimate traffic spikes and distributed denial-of-service attacks, automatically adjusting mitigation strategies to maintain service availability while blocking malicious flows [26]. These systems outperform static rule-based firewalls in scenarios involving evolving attack patterns and complex application-layer protocols [27].

Multi-agent AC approaches have been proposed to address distributed threat detection and response in large-scale enterprise networks where centralized control becomes computationally infeasible. Cooperative multi-agent RL frameworks enable multiple AC agents deployed across different network segments to share threat intelligence and coordinate defensive actions [28]. Studies show that decentralized AC agents can learn emergent collaborative behaviors such as distributed attack tracing, coordinated traffic redirection, and load balancing of security monitoring tasks [29]. The communication overhead and coordination mechanisms required for effective multi-agent AC systems remain active research topics, with recent work exploring attention mechanisms and graph neural networks for inter-agent message passing [30].

Transfer learning and meta-learning approaches have been investigated to address the sample efficiency challenges inherent in applying AC methods to cybersecurity domains where gathering training experience may compromise security. Researchers have demonstrated that AC policies trained in simulated cyber ranges can be fine-tuned for deployment in production networks with significantly reduced training data requirements [31]. Meta-learning frameworks that train AC agents to quickly adapt to new attack types show promise for handling zero-day threats and rapidly evolving malware families [32]. Domain randomization techniques applied during training improve the robustness and generalization of learned policies when transitioning from simulation to real network environments [33].

Adversarial robustness of AC-based security systems has emerged as a critical concern, as attackers may attempt to manipulate the learning process or exploit vulnerabilities in trained policies. Research has examined adversarial attacks against the state observations provided to AC agents, including perturbations to network traffic features and manipulated system logs designed to cause misclassification [13]. Defense mechanisms including adversarial training, certified robustness methods, and ensemble approaches have been proposed to harden AC-based threat detection systems against such attacks [34]. Studies indicate that carefully designed reward functions and architectural choices can improve resilience to adversarial manipulation, though guaranteed robustness remains an open challenge [35].

Explainability and interpretability of AC-based security systems present ongoing challenges for deployment in regulated industries and safety-critical applications. The black-box nature of deep neural networks underlying actor and critic networks complicates security audit requirements and makes it difficult for analysts to understand why specific defensive actions were recommended [12]. Recent research has explored attention mechanisms, saliency maps, and counterfactual explanation techniques to provide insights into AC decision-making processes [36]. Some studies propose hybrid architectures that combine AC learning with symbolic reasoning systems to generate human-readable justifications for automated security decisions [37].

ISSN: 1813-4890

Benchmark datasets and evaluation methodologies for AC-based cybersecurity systems have received increased attention as the field matures. Researchers have identified limitations in existing intrusion detection datasets that may lead to overly optimistic performance estimates and poor generalization to real-world deployments [38]. Efforts to develop more realistic cyber range environments for training and testing AC agents include the integration of sophisticated attacker simulation, realistic background traffic generation, and diverse network topologies representative of enterprise infrastructure [39]. Standardized evaluation protocols considering not only detection accuracy but also response effectiveness, false positive costs, and adaptability to evolving threats have been proposed to facilitate meaningful comparisons across different AC approaches [40].

The computational requirements for training and deploying AC-based security systems in operational environments represent practical considerations addressed in recent literature. Studies have examined the trade-offs between model complexity, inference latency, and detection performance, particularly for inline security applications where processing delays directly impact network performance [41]. Techniques including model compression, quantization, and hardware acceleration using GPUs or specialized AI chips have been investigated to enable real-time AC-based threat detection at network line rates [42]. Edge deployment of lightweight AC agents for distributed threat detection in IoT environments presents additional constraints on model size and computational resources [43].

3. Actor-Critic Reinforcement Learning Fundamentals

The AC framework represents a hybrid approach in RL that combines the strengths of policy-based and value-based methods to achieve efficient and stable learning in complex environments. At its core, the AC architecture consists of two interconnected components: the actor, which learns a policy mapping states to actions, and the critic, which estimates the value function to evaluate the quality of the actor's decisions [3]. This dual-network structure addresses fundamental challenges in pure policy gradient methods, specifically the high variance of gradient estimates that can lead to unstable learning and slow convergence [7]. The critic provides a baseline for evaluating actions, effectively reducing variance while maintaining the bias properties necessary for convergent learning.

The theoretical foundation of AC methods builds upon the policy gradient theorem, which provides a framework for directly optimizing parameterized policies through gradient ascent on expected cumulative rewards. The policy gradient can be expressed in terms of the advantage function, which measures how much better an action is compared to the average action in a given state [44]. The actor updates its parameters in the direction of the policy gradient, estimated using advantage values provided by the critic. The critic simultaneously learns to approximate the value function through temporal difference learning, using observed rewards and value estimates of subsequent states to generate training targets [45]. This bootstrapping approach enables learning from incomplete episodes and supports continuous online learning in environments without natural episode boundaries, characteristics particularly relevant to cybersecurity applications where threats persist indefinitely.

The A2C algorithm implements this framework using separate neural networks for the actor and critic, with both networks typically sharing lower-level feature extraction layers to improve sample efficiency and enable transfer of learned representations [46]. The A2C algorithm employs synchronous parallel actors that interact with multiple environment instances, collecting experiences that are used to compute gradient estimates with reduced variance due to averaging across parallel samples [5]. This synchronization provides more stable training compared to purely asynchronous approaches while maintaining computational efficiency through parallelization. In the context of threat detection, parallel actors can simultaneously

monitor different network segments or analyze multiple data streams, aggregating their experiences to learn a unified defensive policy.

The A3C extends the A2C framework by allowing multiple actors to update the shared policy parameters asynchronously, eliminating the synchronization overhead and enabling more efficient utilization of computational resources [5]. Each A3C worker independently interacts with its environment copy, computes local gradient estimates, and applies them to the global policy network without waiting for other workers to complete their updates. This asynchronous approach can lead to more diverse exploration as different workers may be at various stages of an episode simultaneously, potentially discovering different aspects of optimal defensive strategies. However, the lack of synchronization can introduce some instability in learning, particularly when gradient updates from different workers conflict due to rapid policy changes [47].

SAC introduces an entropy regularization term to the standard RL objective, encouraging the learned policy to maintain high entropy and thus explore a diverse range of actions even after substantial training [6]. This maximum entropy framework provides several benefits for cybersecurity applications, including improved robustness to model misspecification, better exploration of the defensive action space, and more stable learning through automatic temperature tuning that balances exploration and exploitation. The SAC algorithm employs a stochastic policy parameterized by a neural network that outputs action distribution parameters, typically mean and variance for continuous action spaces. The critic is implemented as a pair of Q-function networks that provide ensemble estimates of action values, with the minimum used for gradient computation to reduce overestimation bias [48]. This twin critic architecture significantly improves learning stability and final policy performance compared to single critic approaches.

DDPG represents an AC method specifically designed for continuous control problems, employing a deterministic policy that directly outputs action values rather than probability distributions [5]. DDPG combines insights from deterministic policy gradient theory with deep Q-network techniques, including experience replay and target networks, to enable stable learning with neural function approximators. The algorithm maintains four networks: an actor network, a critic network, and corresponding target networks that provide stable training targets by being updated slowly through soft updates [49]. Experience replay allows the agent to learn from past experiences multiple times, improving sample efficiency, a critical consideration for cybersecurity applications where gathering training data may be expensive or risky. The deterministic nature of DDPG policies can be advantageous for security operations requiring predictable and consistent responses to specific threat scenarios.

Value function approximation in AC methods typically employs temporal difference learning with bootstrapping, where the critic learns to predict expected cumulative rewards from each state by minimizing the squared temporal difference error [45]. The TD error represents the difference between the current value estimate and the target value computed from the observed reward plus the discounted value estimate of the next state. This bootstrapping approach enables learning from incomplete trajectories and facilitates online learning, but introduces bias when function approximation is employed, particularly with neural networks. Advanced AC algorithms address this bias-variance tradeoff through techniques such as generalized advantage estimation, which provides a spectrum of estimators interpolating between high-bias low-variance and low-bias high-variance extremes [44].

Off-policy AC algorithms such as SAC and DDPG learn from experiences collected under different behavior policies than the current policy being optimized, enabling the use of experience replay buffers that store past transitions for repeated sampling [6]. This off-policy capability significantly improves sample efficiency by reusing past experiences multiple times and enables learning from demonstrations or offline datasets collected during previous

ISSN: 1813-4890

security operations. However, off-policy learning introduces additional challenges related to distribution shift between the behavior policy that generated the data and the target policy being learned. Importance sampling and related techniques correct for this distribution mismatch, though they can increase gradient variance and complexity [50].

The integration of recurrent architectures into AC methods enables processing of partially observable environments where the current observation does not contain complete information about the underlying state [20]. Recurrent AC algorithms employ LSTM or GRU networks to maintain internal memory of past observations, enabling the agent to make decisions based on historical context. This capability is essential for detecting sophisticated attacks that manifest through temporal patterns across multiple observations, such as APT campaigns involving reconnaissance followed by exploitation and lateral movement. The recurrent critic can evaluate the quality of actions in the context of the historical trajectory, providing more informative learning signals for the actor compared to critics operating only on current observations.

Hierarchical AC architectures extend the basic framework to learn policies at multiple temporal and spatial abstractions, enabling agents to reason about both high-level strategic objectives and low-level tactical actions [51]. A high-level AC module selects abstract goals or subtasks, while lower-level AC modules learn policies to achieve those goals. This hierarchical decomposition can significantly improve learning efficiency in complex cybersecurity scenarios involving multiple stages of threat detection, analysis, and response. For example, a high-level actor might select between different defensive strategies such as active monitoring, threat hunting, or system isolation, while low-level actors implement the specific actions required for each strategy.

Figure 1: Actor-Critic Architecture for Autonomous Threat Detection

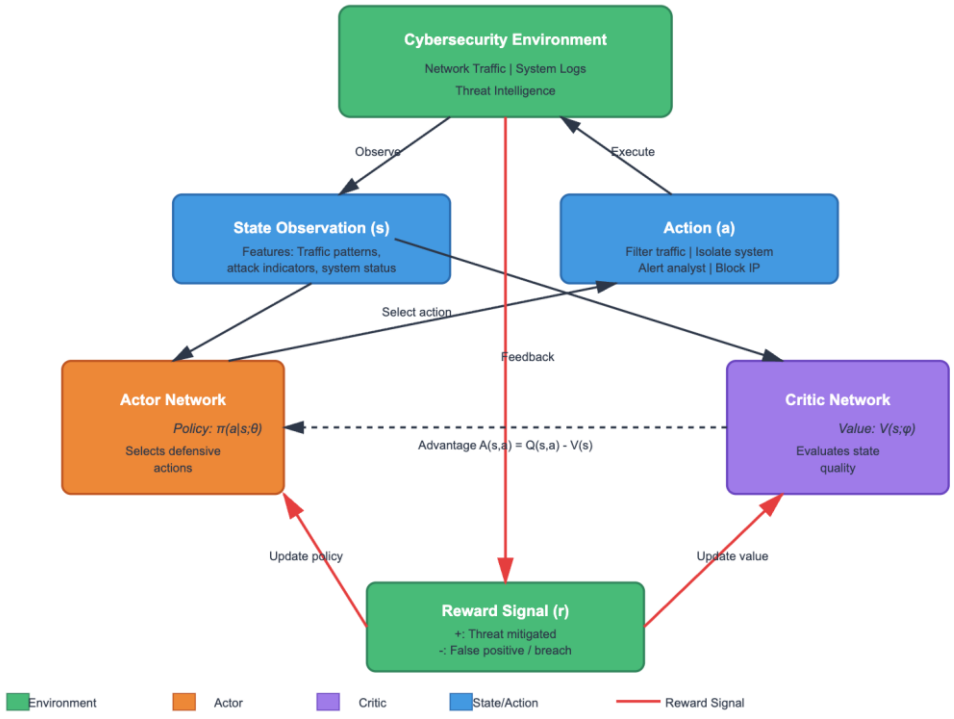

Figure 1. Schematic diagram of the actor-critic reinforcement learning architecture for autonomous threat detection and response. The system consists of two neural networks: the actor network π(a|s;θ) that learns to select defensive actions, and the critic network V(s;φ) that

ISSN: 1813-4890

evaluates state quality*. The cybersecurity environment provides state observations (network traffic features, system logs, threat intelligence) to both networks. The actor selects defensive actions (traffic filtering, system isolation, IP blocking)* which *are executed in the environment. The environment returns a reward signal based on threat mitigation effectiveness and operational impact (false positives). The critic computes advantage values A(s,a) to guide actor policy updates, reducing variance in gradient estimates and enabling stable learning of optimal defensive strategies.*

4. Applications in Threat Detection

AC reinforcement learning has been extensively applied to network intrusion detection, where the agent learns to classify network traffic or system events as benign or malicious based on observed features and patterns. The AC framework naturally accommodates the sequential nature of network traffic analysis, where decisions about current packets or flows may depend on historical context and where early detection of multi-stage attacks requires maintaining awareness of prior suspicious activities [16]. The state representation typically includes statistical features extracted from network packets such as packet size distributions, inter-arrival times, protocol types, connection patterns, and payload characteristics. High-dimensional raw packet data can be processed through convolutional neural network layers that serve as feature extractors feeding into the actor and critic networks [8].

The action space for AC-based intrusion detection systems ranges from binary classification decisions on individual packets to more complex responses involving confidence scores, threat severity assessments, and recommended mitigation actions. Continuous action spaces enable the agent to output probability distributions over potential threat categories, allowing security analysts to prioritize investigations based on detection confidence levels [17]. Some AC implementations learn multi-action policies that simultaneously predict attack types and select appropriate response strategies, unifying detection and response into a single learning framework. The reward function design critically influences learning effectiveness, typically incorporating terms for correct threat identification, penalties for false positives that disrupt legitimate operations, and bonuses for early detection that enables rapid response before significant damage occurs [10].

Experimental evaluations on standard intrusion detection benchmarks demonstrate that AC methods achieve competitive or superior performance compared to traditional machine learning approaches and deep learning classifiers [16]. On the NSL-KDD dataset, AC-based detectors have reported accuracy exceeding 96% with false positive rates below 3%, representing improvements of several percentage points over random forest and support vector machine baselines. More importantly, AC agents demonstrate better adaptability to evolving attack patterns through continued learning, whereas static classifiers require retraining on new labeled data to maintain detection performance against novel threats [17]. The ability to fine-tune policies based on operational feedback enables AC systems to customize their behavior to specific network environments and organizational security policies.

Zero-day attack detection represents a particularly challenging application where AC methods offer advantages over supervised learning approaches that rely on labeled examples of known attacks. The RL paradigm enables agents to learn general principles of anomalous behavior rather than memorizing signatures of specific attacks, potentially identifying novel threat patterns that differ from training examples [1]. AC agents trained to maximize long-term security objectives while minimizing disruption to normal operations can discover zero-day attacks through anomaly detection based on deviations from learned models of legitimate traffic patterns. Reward shaping techniques that penalize unrecognized behaviors encourage

ISSN: 1813-4890

cautious policies that investigate suspicious activities even when they do not match known attack signatures [52].

Phishing and social engineering attack detection has benefited from AC methods applied to email content analysis and user behavior monitoring. The agent learns to identify linguistic patterns, sender characteristics, and contextual features indicative of phishing attempts by receiving rewards for correct identification and penalties for failing to block malicious messages or incorrectly filtering legitimate communications [53]. Time-series analysis of user email interactions enables the AC system to detect anomalous patterns such as unexpected requests from trusted contacts whose accounts may have been compromised. The continuous learning capability allows the system to adapt to evolving phishing tactics, including personalized spear-phishing campaigns that target specific individuals with tailored social engineering content.

Malware detection and classification using AC reinforcement learning focuses on behavioral analysis rather than static code signatures, providing robustness against obfuscation techniques and polymorphic malware [21]. The agent monitors program execution behaviors including system calls, file system operations, network communications, and registry modifications, learning to distinguish malicious activities from legitimate software behaviors. The state representation encodes sequences of behavioral events, enabling the AC system to detect malware through patterns of actions rather than individual suspicious operations that might appear benign in isolation [22]. Dynamic analysis environments allow the AC agent to observe malware execution in sandboxed systems, gathering behavioral data while minimizing risk to production infrastructure.

APT detection leveraging AC methods addresses the challenge of identifying coordinated, multi-stage attack campaigns that unfold over extended periods. The agent maintains long-term memory of suspicious activities across the network, learning to correlate seemingly unrelated events that collectively indicate APT presence [18]. Recurrent neural networks in the actor and critic architectures enable processing of temporally extended sequences of indicators of compromise, recognizing patterns such as initial reconnaissance, vulnerability exploitation, privilege escalation, lateral movement, and data exfiltration [19]. The reward function incorporates delayed rewards that provide feedback only after full attack chains are identified, encouraging the agent to reason about long-term consequences of detection decisions rather than optimizing for immediate classification accuracy [20].

Insider threat detection represents another domain where AC reinforcement learning provides valuable capabilities for identifying malicious activities by authorized users with legitimate access to systems and data. The agent learns normal behavior patterns for individual users and roles, detecting deviations that may indicate compromised credentials or malicious insiders [54]. The state representation includes features such as access patterns, data transfers, authentication events, and working hour analysis. The AC framework supports fine-grained anomaly detection that considers the specific context of user actions, distinguishing between legitimate unusual behaviors and genuinely suspicious activities. Reward shaping balances the need to detect insider threats against the risk of false accusations that could damage employee morale and organizational culture.

Cloud environment security monitoring using AC methods addresses challenges specific to virtualized infrastructure, including dynamic resource allocation, multi-tenancy concerns, and limited visibility into underlying physical infrastructure [55]. The agent learns to detect cloud-specific attack vectors such as VM escape attempts, side-channel attacks, and resource abuse. The state representation captures cloud-specific metrics including virtual machine behavior, API call patterns, and resource consumption. The AC system adapts to the elasticity of cloud environments, learning to distinguish between legitimate scaling events and malicious resource exhaustion attacks. Integration with cloud security posture management tools enables

ISSN: 1813-4890

the AC agent to incorporate configuration compliance information into its threat detection decisions [56].

IoT security applications of AC reinforcement learning focus on lightweight models suitable for resource-constrained devices while maintaining effective threat detection capabilities [43]. The agent monitors IoT device communications and behaviors, identifying compromised devices participating in botnets or serving as entry points for network intrusions. The state representation emphasizes communication patterns and protocol behaviors rather than content analysis, reducing computational requirements. Federated learning approaches enable multiple AC agents deployed across IoT devices to collaboratively learn threat detection policies while preserving privacy and reducing communication overhead. The learned policies must balance security with operational constraints such as battery life and bandwidth limitations inherent to IoT environments.

Table 1: Comparison of Actor-Critic Based Intrusion Detection Systems

Study (Year)	Algorithm	Dataset	Accuracy (%)	FPR (%)	Training Time	Key Innovation
Ferrag et al. (2020) [16]	A2C	NSL-KDD	96.2	2.8	3.5 hrs	Parallel experience collection with shared layers
Liu & Lang (2019) [17]	A3C	CICIDS2017	95.8	3.1	4.2 hrs	Asynchronous updates for continuous learning
Christodorescu et al. (2020) [21]	SAC	Malware Dataset	97.1	1.9	6.5 hrs	Entropy regularization for polymorphic malware
Anderson et al. (2020) [22]	SAC	CICIDS2017	96.8	2.4	5.1 hrs	Stochastic policies for adversarial robustness
Doriguzzi-Corin et al. (2020) [27]	DDPG	CICDDoS2019	95.3	2.1	3.9 hrs	Real-time DDoS mitigation with continuous actions
Ghafir et al. (2020) [19]	A3C+LSTM	APT Dataset	92.4	2.9	8.3 hrs	Temporal dependencies for multi-stage attacks
Han et al. (2021) [8]	SAC	UNSW-NB15	96.4	2.6	5.7 hrs	Adversarial training for evasion attack defense

Table 1. Performance comparison of *actor-critic based intrusion detection systems on standard benchmark datasets. SAC-based approaches achieve highest accuracy (96.8-97.1%) and lowest false positive rates (1.9-2.6%) due to entropy* regularization*. A3C methods excel in temporal attack detection when combined with recurrent architectures. DDPG algorithms perform well in continuous control scenarios. Training times range from 3.5 to 8.3 hours on NVIDIA Tesla V100 GPU. All AC-based systems outperform traditional machine learning baselines (random forest, SVM) which typically achieve 85-92% accuracy with 5-8% FPR on same datasets.*

5. Response Mechanisms and Automation

Automated incident response represents a critical application domain for AC reinforcement learning, where agents learn to select and execute appropriate remediation actions in response to detected threats. The AC framework enables learning of complex response policies that consider multiple factors including attack severity, asset criticality, business impact, available defensive resources, and potential collateral damage from aggressive countermeasures [23]. The state representation for response automation typically includes threat intelligence about the detected attack, contextual information about affected systems, current network status, and historical data about previous incident response outcomes. This comprehensive state enables the AC agent to make informed decisions that balance security effectiveness against operational continuity requirements.

The action space for automated response encompasses a wide range of defensive measures organized hierarchically from passive monitoring to aggressive isolation. Low-severity actions include increasing logging verbosity, enabling enhanced monitoring for affected systems, and

alerting security analysts for manual investigation [24]. Medium-severity responses involve targeted traffic filtering, rate limiting suspicious connections, and temporary account lockouts. High-severity actions include complete network isolation of compromised systems, forced termination of malicious processes, and initiation of forensic data collection procedures. The AC agent learns a policy that selects appropriate response actions based on the specific threat context, avoiding both under-response that leaves systems vulnerable and over-response that disrupts legitimate business operations unnecessarily [25].

Reward function design for response automation must carefully balance competing objectives to ensure the learned policy aligns with organizational security and operational goals. Positive rewards accrue for successful threat mitigation, measured through metrics such as reduction in attacker dwell time, prevention of data exfiltration, and containment of malware spread [23]. Negative rewards penalize false positives that disrupt legitimate operations, excessive response actions that unnecessarily impact availability, and slow response times that allow attacks to progress. The reward function often incorporates domain-specific cost models that quantify the business impact of security incidents and response actions, enabling the AC agent to learn policies that optimize security outcomes while considering economic constraints [10]. Multi-objective reward formulations allow balancing of security, availability, and performance considerations through weighted combinations or Pareto optimization approaches.

Security orchestration integration enables AC-based response automation to interface with existing security tools and infrastructure through standardized APIs and playbook frameworks [15]. The actor network outputs abstract response strategies that are translated into concrete actions by orchestration platforms such as SOAR systems. This layered architecture separates high-level policy learning from low-level execution details, improving portability across different security tool ecosystems. The AC agent learns to sequence complex multi-step response procedures, coordinating actions across diverse security tools including firewalls, endpoint detection and response systems, security information and event management platforms, and threat intelligence feeds. Learned orchestration policies can adapt to dynamic conditions such as tool failures or resource constraints, selecting alternative response paths to achieve security objectives.

Dynamic defense strategies learned through AC reinforcement learning go beyond reactive response to detected threats, incorporating proactive and deceptive elements that increase attacker costs and gather intelligence about adversary tactics [4]. The agent learns when to deploy honeypots or deception technologies that redirect attackers to monitored environments where their tools and techniques can be studied safely. AC-based systems can dynamically adjust network topology and service configurations to present attackers with a moving target, invalidating reconnaissance information and disrupting automated attack tools. These adaptive defense mechanisms learn to balance the operational overhead of frequent security posture changes against the security benefits of reducing attacker success rates and extending dwell time for detection [57].

Patch management and vulnerability remediation represent another application area where AC methods enable intelligent automation of security maintenance tasks. The agent learns policies for prioritizing patches based on vulnerability severity, exploit availability, asset criticality, and organizational risk tolerance [58]. The state representation includes vulnerability scan results, threat intelligence about active exploits, system configuration information, and maintenance schedules. The AC system learns to balance security imperatives against operational constraints such as maintenance windows, application compatibility requirements, and change management processes. By learning from outcomes of previous patching decisions, the agent improves its ability to predict which vulnerabilities pose the greatest risk in specific organizational contexts and should receive immediate attention.

ISSN: 1813-4890

Access control policy optimization using AC reinforcement learning enables dynamic adjustment of permissions based on observed user behaviors and threat conditions. The agent learns to tighten access controls in response to suspicious activities or elevated threat levels while relaxing restrictions when appropriate to avoid impeding legitimate operations [54]. The state representation captures current access policies, user access patterns, authentication events, and contextual factors such as user location and device security posture. The AC system learns policies that implement principles of least privilege and zero trust architecture through continuous refinement of access permissions based on ongoing risk assessment. This dynamic approach to access control contrasts with static role-based access control systems that cannot adapt to changing risk conditions or user behavior patterns.

Network traffic management and filtering policies learned through AC methods enable adaptive defense against distributed denial-of-service attacks and malicious traffic flows. The agent learns to distinguish between legitimate traffic spikes caused by flash crowds or viral content and DDoS attacks orchestrated by botnets [26]. The action space includes rate limiting parameters, traffic redirection policies, and connection filtering rules that the AC system dynamically optimizes based on current traffic patterns and attack indicators. Learned policies balance the competing objectives of maintaining service availability for legitimate users while blocking attack traffic, adapting mitigation strategies as attacks evolve in real-time [27]. The AC framework enables learning of sophisticated traffic management strategies that consider application-layer protocol behaviors and client interaction patterns beyond simple volumetric filtering.

Automated threat hunting represents an emerging application where AC agents learn to proactively search for indicators of compromise and undiscovered threats within enterprise networks. The agent selects investigation targets, analysis techniques, and data sources to examine based on threat intelligence, historical incident data, and patterns of suspicious activities [59]. The reward function provides feedback based on the value of threats discovered relative to the investigation effort expended, encouraging efficient allocation of limited security analyst time and computational resources. AC-based threat hunting systems learn to recognize subtle anomalies and correlations that may indicate sophisticated attacks missed by automated detection systems. The learned hunting policies complement reactive detection mechanisms by proactively seeking threats before they trigger alerts or cause damage.

Incident response playbook generation through AC reinforcement learning enables automated discovery of effective response procedures tailored to specific organizational contexts and threat scenarios. Rather than relying solely on generic industry best practice playbooks, the AC agent learns response sequences that perform well in the specific technical and operational environment of the deploying organization [15]. The agent explores different combinations and orderings of response actions, learning through experience which procedures most effectively mitigate various threat types while minimizing operational disruption. The learned playbooks can be reviewed and validated by human security experts before deployment, combining the exploratory power of RL with human domain expertise. Over time, the AC system continuously refines response playbooks based on feedback from actual incident outcomes, improving effectiveness as the threat landscape evolves.

ISSN: 1813-4890

Figure 2: Automated Incident Response Process using Actor-Critic RL

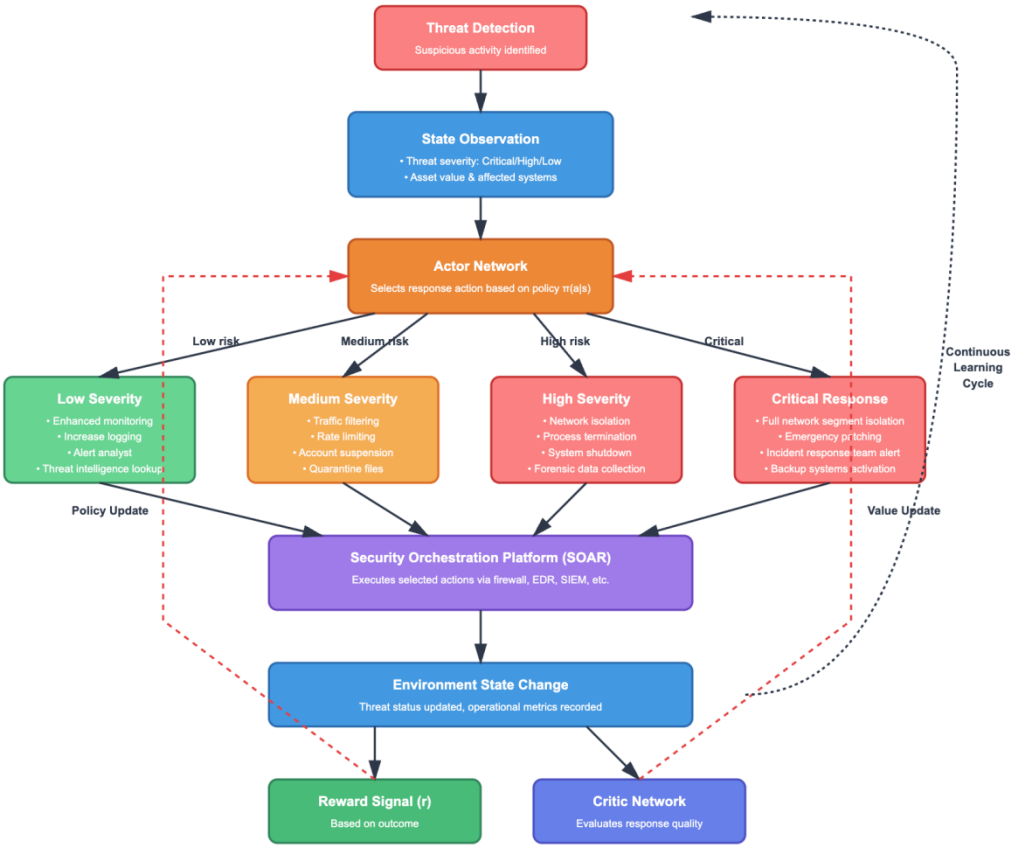

Figure 2. Flowchart depicting the automated incident response process using actor-critic reinforcement learning. When a threat is detected, the system generates a state observation including threat severity, asset criticality, and system status. The actor network selects appropriate response actions based on learned policy π(a|s), ranging from low-severity monitoring to critical system isolation. Actions are categorized by severity: low-severity responses include enhanced monitoring and logging; medium-severity responses involve traffic filtering and account suspension; high-severity responses include network isolation and *process termination; critical responses trigger full segment isolation and incident* response *team activation. Selected actions are executed through security orchestration platforms (SOAR) that interface with firewalls, endpoint detection and response (EDR) systems, and security information and event management (SIEM) platforms. The environment state changes based on executed actions, generating reward signals that reflect attack mitigation effectiveness and operational impact. The critic network evaluates response quality, and both actor and critic update their parameters through feedback loops, enabling continuous improvement of response policies.*

6. Challenges and Future Directions

Sample efficiency remains a fundamental challenge for deploying AC reinforcement learning in production cybersecurity environments, where gathering sufficient training experience through trial-and-error experimentation poses unacceptable security risks. Unlike simulation-based domains such as game playing or robotics where failures during training carry minimal consequences, errors by learning security systems can result in successful attacks, data breaches, or disruption of critical services [14]. The number of environment interactions required to train effective AC policies often ranges from thousands to millions of episodes, far exceeding what can be safely collected in live operational networks. Transfer

learning approaches that train agents in simulated environments and fine-tune them with limited real-world experience show promise but struggle with the reality gap between simulations and production systems [31]. Offline RL methods that learn from historical security logs and incident response records without requiring interactive exploration offer potential solutions but must address distribution shift challenges and inability to discover novel strategies not represented in historical data [50].

Interpretability and explainability of AC-based security systems present critical challenges for deployment in regulated industries and safety-critical applications where human oversight and audit trails are mandatory. The deep neural networks underlying actor and critic components function as black boxes whose decision-making processes are opaque to security analysts and compliance auditors [12]. Understanding why an AC agent recommended a specific response action or assigned a particular threat severity score becomes difficult when the policy is represented by millions of neural network parameters. This opacity complicates troubleshooting of system failures, validation of correct behavior, and building trust with security operations teams who must rely on automated decisions. Explainable RL techniques including attention mechanisms, saliency analysis, and counterfactual reasoning provide partial solutions but often sacrifice model performance or provide explanations that remain too abstract for actionable operational insights [36].

Adversarial attacks against AC-based security systems represent an emerging threat vector where sophisticated attackers exploit vulnerabilities in the learning process or trained policies to evade detection or manipulate defensive responses. State observation poisoning attacks inject crafted malicious traffic or false telemetry data designed to cause misclassification by confusing the AC agent's perception of the environment [13]. Policy manipulation attacks during training can subtly influence reward signals or state transitions to teach the agent exploitable weaknesses. Adversarial examples that cause trained AC detectors to misclassify obvious attacks have been demonstrated in research settings, raising concerns about robustness against adaptive adversaries [34]. Defensive techniques including adversarial training, certified robustness methods, and ensemble diversity provide some protection but cannot eliminate all vulnerabilities, particularly against adaptive attackers with knowledge of the defense mechanisms [35].

The reality gap between training environments and deployment contexts creates substantial challenges for AC systems developed using simulation-based approaches. Simulated network traffic, attack behaviors, and system dynamics inevitably fail to capture the full complexity and variability of production environments [11]. AC agents trained in simplistic simulations may learn brittle policies that exploit simulator artifacts rather than developing robust defensive strategies that generalize to real-world conditions. Even sophisticated cyber range environments struggle to replicate factors such as legitimate user behavior diversity, application-specific protocols, organizational workflows, and subtle attack techniques employed by skilled human adversaries [39]. Domain randomization techniques that expose agents to diverse simulated conditions during training improve robustness but cannot eliminate all aspects of the reality gap, particularly for rare but critical scenarios underrepresented in training data [33].

Scalability challenges arise when deploying AC-based security systems in large enterprise networks with heterogeneous devices, diverse applications, and massive data volumes requiring analysis. Training centralized AC policies on aggregated data from thousands of endpoints and network segments creates computational bottlenecks and privacy concerns [41]. Distributed training approaches using multiple parallel actors help but face challenges related to communication overhead, synchronization delays, and ensuring consistent policy updates across geographically dispersed infrastructure [28]. Federated learning architectures that train local AC agents on individual network segments and aggregate their learned policies show

promise for preserving privacy while enabling collaborative learning [29]. However, federated approaches must address challenges including non-IID data distributions across sites, handling of stragglers with slow computation or communication, and robustness against malicious participants that could poison the aggregated policy.

Integration with human security analysts represents both a technical challenge and an opportunity for AC-based automation systems. Fully autonomous security operations without human oversight raise concerns about accountability, unintended consequences, and inability to handle novel scenarios outside the agent's training distribution [12]. Human-in-the-loop approaches that combine AC automation with human judgment leverage complementary strengths: agents handle routine high-volume tasks while humans provide oversight for critical decisions and handle complex edge cases. Designing effective human-agent teaming requires careful consideration of factors including appropriate automation levels, transparency of agent reasoning, mechanisms for human intervention and override, and workload management to prevent alert fatigue [60]. Active learning frameworks where AC agents identify ambiguous cases for human labeling can improve policy learning while minimizing analyst burden.

Multi-stage attack detection and response present challenges related to temporal credit assignment, where the consequences of defensive actions taken early in an attack campaign may not become apparent until much later. AC methods using temporal difference learning bootstrap value estimates from subsequent states, but this bootstrapping can blur credit assignment when rewards are sparse and delayed [45]. Sophisticated attacks that unfold over weeks or months create extremely long time horizons that exacerbate credit assignment difficulties [19]. Hierarchical RL approaches that decompose the problem into subgoals at multiple temporal scales show promise for handling extended attack campaigns [51]. Options frameworks and goal-conditioned policies that enable compositional learning of defensive skills may improve sample efficiency and credit assignment for multi-stage threat scenarios.

Concept drift in attack patterns and normal behavior baselines creates challenges for maintaining AC policy effectiveness over time as the operational environment evolves. Networks undergo continuous changes including new applications, evolving user behaviors, infrastructure upgrades, and shifting threat landscapes [1]. AC agents trained on historical data may become less effective as their training distribution diverges from current operating conditions. Online continual learning approaches that enable policies to adapt to new conditions while avoiding catastrophic forgetting of previously learned defensive strategies represent an active research area [46]. Detecting when AC policies have become stale and require retraining or updating involves monitoring performance metrics and statistical properties of observed data distributions, but determining appropriate thresholds and retraining triggers remains challenging.

Reward function specification and alignment represent fundamental challenges in applying AC methods to complex cybersecurity objectives that resist precise mathematical formulation. Security goals such as protecting sensitive data, maintaining operational resilience, and preserving organizational reputation involve subjective value judgments that cannot be fully captured in simple reward functions [10]. Misspecified rewards lead to reward hacking behaviors where agents discover loopholes that maximize the stated objective while violating the intended security properties. Inverse RL approaches that infer reward functions from expert demonstrations of security analyst decision-making show promise but require substantial high-quality demonstration data [52]. Multi-objective RL formulations that explicitly model trade-offs between competing goals such as security and usability enable more nuanced policy learning but complicate the training process and policy evaluation.

Coordination challenges in multi-agent AC systems deployed across distributed security infrastructure include communication complexity, emergent behaviors, and handling of conflicting objectives between agents. Decentralized AC agents operating on different network

segments must share information about threats and coordinate defensive responses without overwhelming communication channels or creating exploitable coordination protocols [28]. Emergent phenomena in multi-agent systems can lead to unstable oscillations, deadlocks, or other unintended collective behaviors that compromise security [30]. Game-theoretic frameworks that model interactions between defensive agents and adversaries as competitive games provide theoretical grounding for multi-agent security systems but often require simplifying assumptions about opponent capabilities and rationality that may not hold for sophisticated attackers.

Future research directions include the development of meta-learning approaches that enable AC agents to rapidly adapt to novel attack types with minimal additional training data. Meta-RL methods train agents to learn efficiently by exposing them to diverse tasks during meta-training, developing learning algorithms rather than fixed policies [32]. Applied to cybersecurity, meta-learning could enable rapid adaptation to zero-day exploits and emerging threat categories through few-shot learning from limited examples. Model-agnostic meta-learning and other gradient-based meta-learning approaches show promise for this application but require substantial computational resources and careful design of meta-training task distributions.

Causal reasoning and counterfactual analysis integrated with AC architectures represent promising directions for improving robustness and interpretability of security automation systems. Causal models that explicitly represent relationships between attacker actions, defensive responses, and security outcomes enable agents to reason about intervention effects rather than relying solely on correlational patterns [37]. Counterfactual explanations that describe how alternative actions would have changed outcomes provide more actionable insights for security analysts compared to standard attribution methods [36]. Structural causal model integration with deep RL remains an active research area with potential to improve generalization and enable reasoning under distribution shift.

Hybrid neuro-symbolic approaches that combine AC learning with logical reasoning systems may address interpretability challenges while retaining the learning efficiency of neural network function approximators. Symbolic components can encode security policies, compliance requirements, and domain knowledge in human-readable logical rules, while neural components handle high-dimensional perception and pattern recognition [37]. Recent work on differentiable logic and neural theorem proving enables end-to-end learning of hybrid systems that combine symbolic and subsymbolic reasoning. Applied to cybersecurity, such hybrid architectures could learn to detect attacks while providing formal verification of policy compliance and interpretable explanations grounded in symbolic security rules.

7. Conclusion

AC reinforcement learning has emerged as a powerful paradigm for developing autonomous threat detection and response systems that can adapt to evolving cybersecurity challenges through continuous learning from experience. The AC framework addresses fundamental limitations of traditional security approaches by enabling agents to discover optimal defensive strategies without exhaustive labeled datasets or predefined rules, learning instead through interaction with the operational environment and feedback from security outcomes. The dual-network architecture combining policy learning and value function approximation provides stable and efficient training for complex sequential decision-making tasks characteristic of modern security operations. Advanced AC variants including A2C, A3C, SAC, and DDPG demonstrate strong performance across diverse cybersecurity applications ranging from network intrusion detection to automated incident response, achieving detection accuracies exceeding conventional methods while providing adaptability to novel threats.

The application of AC methods to threat detection has yielded systems capable of identifying sophisticated attack patterns including APT campaigns, zero-day exploits, and polymorphic malware through behavioral analysis and temporal reasoning. These systems process high-dimensional security telemetry encompassing network traffic, system logs, and threat intelligence feeds, learning to distinguish malicious activities from benign operations while minimizing false positive rates that plague traditional detection tools. Recurrent architectures enable AC agents to maintain awareness of historical context, recognizing coordinated multi-stage attacks that unfold over extended periods. The continuous learning capability allows deployed systems to refine detection policies based on operational feedback, adapting to changing threat landscapes and organizational environments without requiring complete retraining.

Automated response mechanisms powered by AC algorithms demonstrate the potential for intelligent orchestration of defensive actions that balance security effectiveness against operational constraints. Learned response policies consider multiple factors including threat severity, asset criticality, and business impact when selecting mitigation actions, achieving faster and more consistent incident response compared to manual procedures. Dynamic defense strategies incorporating deception, moving target defense, and adaptive access control leverage the sequential decision-making capabilities of AC methods to implement sophisticated defensive maneuvers that increase attacker costs and improve security postures. Integration with security orchestration platforms enables AC agents to coordinate actions across diverse security tools, translating high-level strategic decisions into concrete defensive measures.

Despite promising advances, significant challenges remain before AC-based security systems achieve widespread production deployment. Sample efficiency constraints require development of training methodologies that learn effective policies without exposing production systems to unacceptable risks during exploration. The reality gap between simulated training environments and operational networks necessitates robust transfer learning approaches and sim-to-real techniques that preserve policy effectiveness across domain shifts. Adversarial robustness against attackers who may attempt to manipulate learning processes or exploit policy vulnerabilities demands careful design of defensive mechanisms and continuous monitoring for adversarial perturbations. Interpretability limitations complicate building trust with security analysts and meeting regulatory requirements for explainable automated decisions in safety-critical contexts.

Scalability considerations for deploying AC systems in large enterprise environments with heterogeneous infrastructure and massive data volumes require distributed architectures that balance computational efficiency with learning effectiveness. Coordination between multiple AC agents operating across different network segments presents challenges related to communication overhead, emergent behaviors, and alignment of local and global security objectives. Integration of human analysts into the automation loop requires thoughtful interface design that enables effective collaboration between human expertise and agent capabilities while avoiding alert fatigue and maintaining appropriate oversight. Long-term deployment challenges including concept drift, reward misspecification, and maintaining policy effectiveness as environments evolve necessitate ongoing research into continual learning and adaptive systems.

Future development of AC-based security automation will likely focus on meta-learning approaches enabling rapid adaptation to novel threats, hybrid neuro-symbolic architectures combining learning efficiency with interpretability, and multi-agent coordination frameworks for distributed defense. Advances in offline and batch RL may enable learning from historical security data without requiring risky online exploration, while improved simulation fidelity through high-fidelity cyber ranges could narrow the reality gap. Causal reasoning capabilities integrated with AC methods may enhance robustness and enable counterfactual analysis

supporting better security decisions. As these research directions mature and practical deployment challenges are addressed through engineering innovation, AC reinforcement learning will increasingly realize its potential to transform cybersecurity operations through intelligent automation that complements and augments human security expertise.

References

[1] Apruzzese, G., Colajanni, M., Ferretti, L., & Marchetti, M. (2019, May). Addressing adversarial attacks against security systems based on machine learning. In 2019 11th international conference on cyber conflict (CyCon) (Vol. 900, pp. 1-18). IEEE.

[2] IBM Security, M. (2023). Cost of a data breach report 2021.

[3] Nguyen TT, Reddi VJ. Deep reinforcement learning for cyber security. IEEE Trans Neural Netw Learn Syst. 2021;32(8):3779-3795.

[4] Huang L, Zhu Q. Adaptive strategic cyber defense for advanced persistent threats in critical infrastructure networks. ACM SIGMETRICS Perform Eval Rev. 2019;47(2):52-56.

[5] Kumar, H., Koppel, A., & Ribeiro, A. (2023). On the sample complexity of actor-critic method for reinforcement learning with function approximation. Machine Learning, 112(7), 2433-2467.

[6] Li, S., Wu, Y., Cui, X., Dong, H., Fang, F., & Russell, S. (2019, July). Robust multi-agent reinforcement learning via minimax deep deterministic policy gradient. In Proceedings of the AAAI conference on artificial intelligence (Vol. 33, No. 01, pp. 4213-4220).

[7] Ferrag, M. A., Shu, L., Friha, O., & Yang, X. (2021). Cyber security intrusion detection for agriculture 4.0: Machine learning-based solutions, datasets, and future directions. IEEE/CAA Journal of Automatica Sinica, 9(3), 407-436.

[8] Han D, Wang Z, Zhong Y, et al. Evaluating and improving adversarial robustness of machine learning-based network intrusion detectors. IEEE J Sel Areas Commun. 2021;39(8):2632-2647.

[9] Eghtesad, T., Vorobeychik, Y., & Laszka, A. (2020, October). Adversarial deep reinforcement learning based adaptive moving target defense. In International Conference on Decision and Game Theory for Security (pp. 58-79). Cham: Springer International Publishing.

[10] Zheng, K., Sun, Z., Song, Y., Zhang, C., Zhang, C., Chang, F., ... & Fu, X. (2025). Stochastic scenario generation methods for uncertainty in wind and photovoltaic power outputs: A comprehensive review. Energies, 18(3), 503.

[11] Ahmad Z, Shahid Khan A, Wai Shiang C, et al. Network intrusion detection system: A systematic study of machine learning and deep learning approaches. Trans Emerg Telecommun Technol. 2021;32(1):e4150.

[12] Chen, L. P. (2019). Mehryar Mohri, Afshin Rostamizadeh, and Ameet Talwalkar: Foundations of machine learning: The MIT Press, Cambridge, MA, 2018, 504 pp., CDN $96.53 (hardback), ISBN 9780262039406.

[13] Papernot N, McDaniel P, Sinha A, Wellman MP. SoK: Security and privacy in machine learning. IEEE European Symposium on Security and Privacy. IEEE; 2020. p. 399-414.

[14] Zhang, P., Zhang, Z., & Chao, H. C. (2020). A stacked human activity recognition model based on parallel recurrent network and time series evidence theory. Sensors, 20(14), 4016.

[15] Sethi K, Kumar R, Prajapati N, Bera P. Deep reinforcement learning based intrusion detection system for cloud infrastructure. IEEE Int Conf Cloud Comput Technol Sci. 2020;2020:1-6.

[16] Ferrag MA, Maglaras L, Moschoyiannis S, Janicke H. Deep learning for cyber security intrusion detection: Approaches, datasets, and comparative study. J Inf Secur Appl. 2020;50:102419.

[17] Liu H, Lang B. Machine learning and deep learning methods for intrusion detection systems: A survey. Appl Sci. 2019;9(20):4396.

[18] Yusof, Z. B. (2024). Exploration of advanced persistent threats: techniques, mitigation strategies, and impacts on critical infrastructure. International Journal of Advanced Cybersecurity Systems, Technologies, and Applications, 8(12), 1-9.

[19] Basnet, A. S., Ghanem, M. C., Dunsin, D., Kheddar, H., & Sowinski-Mydlarz, W. (2025). Advanced persistent threats (apt) attribution using deep reinforcement learning. Digital Threats: Research and Practice.

[20] Canaan, B. (2023). Microgrid real-time active power diagnostic against cyber-physical attacks using recurrent neural networks (Doctoral dissertation, Université de Haute Alsace-Mulhouse).

[21] Campion, M., Dalla Preda, M., & Giacobazzi, R. (2021). Learning metamorphic malware signatures from samples. Journal of Computer Virology and Hacking Techniques, 17(3), 167-183.

[22] Mohammed, A. S., & Okafor, E. (2025, June). Off-Policy Inspired Imitation Learning for Generation of Adversarial Malware. In IFIP International Conference on Artificial Intelligence Applications and Innovations (pp. 257-269). Cham: Springer Nature Switzerland.

[23] Hasan, K., Shetty, S., Hassanzadeh, A., & Ullah, S. (2019, November). Towards optimal cyber defense remediation in cyber physical systems by balancing operational resilience and strategic risk. In MILCOM 2019-2019 IEEE Military Communications Conference (MILCOM) (pp. 1-8). IEEE.

[24] Hu Z, Beuran R, Tan Y. Automated penetration testing using deep reinforcement learning. IEEE European Symposium on Security and Privacy Workshops. IEEE; 2020. p. 2-10.

[25] Ghanem, M. C., & Chen, T. M. (2019). Reinforcement learning for efficient network penetration testing. Information, 11(1), 6.

[26] Chowdhary A, Pisharody S, Alshamrani A, Huang D. Dynamic game based security framework in SDN-enabled cloud networking environments. ACM Int Workshop Secur Softw Defined Netw Netw Funct Virtualization. 2019;2019:53-58.

[27] Doriguzzi-Corin R, Millar S, Scott-Hayward S, et al. LUCID: A practical, lightweight deep learning solution for DDoS attack detection. IEEE Trans Netw Serv Manag. 2020;17(2):876-889.

[28] Tang, M., & Wong, V. W. (2020). Deep reinforcement learning for task offloading in mobile edge computing systems. IEEE Transactions on Mobile Computing, 21(6), 1985-1997.

[29] Ahmed, I., Syed, M. A., Maaruf, M., & Khalid, M. (2025). Distributed computing in multi-agent systems: a survey of decentralized machine learning approaches. Computing, 107(1), 2.

[30] Liu Y, Wang W, Hu YH, et al. Multi-agent game abstraction via graph attention neural network. AAAI Conference on Artificial Intelligence. AAAI; 2020. p. 7211-7218.

[31] Yoon J, Kim D, Yoon K, et al. Transferable deep reinforcement learning framework for autonomous vehicles with joint sensor fusion. IEEE Trans Intell Transp Syst. 2021;22(4):2304-2314.

[32] Wang JX, King M, Porcel N, et al. Alchemy: A benchmark and analysis toolkit for meta-reinforcement learning agents. Adv Neural Inf Process Syst. 2021;34:3175-3188.

[33] Shakerimov, A., Alizadeh, T., & Varol, H. A. (2023). Efficient sim-to-real transfer in reinforcement learning through domain randomization and domain adaptation. IEEE Access, 11, 136809-136824.

[34] Reda, H. T., Anwar, A., Mahmood, A. N., & Tari, Z. (2023). A taxonomy of cyber defence strategies against false data attacks in smart grids. ACM Computing Surveys, 55(14s), 1-37.

[35] Ilyas A, Santurkar S, Tsipras D, et al. Adversarial examples are not bugs, they are features. Adv Neural Inf Process Syst. 2019;32:125-136.

[36] Li, P., Bahri, O., Hosseinzadeh, P., Boubrahimi, S. F., & Hamdi, S. M. (2024). Info-CELS: Informative Saliency Map Guided Counterfactual Explanation. arXiv preprint arXiv:2410.20539.

[37] Xing, S., Wang, Y., & Liu, W. (2025). Multi-Dimensional Anomaly Detection and Fault Localization in Microservice Architectures: A Dual-Channel Deep Learning Approach with Causal Inference for Intelligent Sensing. Sensors, 25(11), 3396.

[38] Kenyon A, Deka L, Elizondo D. Are public intrusion datasets fit for purpose characterising the state of the art in intrusion event datasets. Comput Secur. 2020;99:102022.

[39] Yamin MM, Katt B, Gkioulos V. Cyber ranges and security testbeds: Scenarios, functions, tools and architecture. Comput Secur. 2020;88:101636.

[40] Aminu, M., Akinsanya, A., Dako, D. A., & Oyedokun, O. (2024). Enhancing cyber threat detection through real-time threat intelligence and adaptive defense mechanisms. International Journal of Computer Applications Technology and Research, 13(8), 11-27.

[41] Ring M, Wunderlich S, Scheuring D, et al. A survey of network-based intrusion detection data sets. Comput Secur. 2019;86:147-167.

[42] Hussain F, Hussain R, Hassan SA, Hossain E. Machine learning in IoT security: Current solutions and future challenges. IEEE Commun Surv Tutor. 2020;22(3):1686-1721.

[43] Mothukuri V, Parizi RM, Pouriyeh S, et al. A survey on security and privacy of federated learning. Future Gener Comput Syst. 2021;115:619-640.

[44] Bennett, D., Davidson, G., & Niv, Y. (2022). A model of mood as integrated advantage. Psychological Review, 129(3), 513.

[45] Ramprasad, P., Li, Y., Yang, Z., Wang, Z., Sun, W. W., & Cheng, G. (2023). Online bootstrap inference for policy evaluation in reinforcement learning. Journal of the American Statistical Association, 118(544), 2901-2914.

[46] Parisi GI, Kemker R, Part JL, et al. Continual lifelong learning with neural networks: A review. Neural Netw. 2019;113:54-71.

[47] Khan, Q. W., Khan, A. N., Rizwan, A., Ahmad, R., Khan, S., & Kim, D. H. (2023). Decentralized machine learning training: a survey on synchronization, consolidation, and topologies. IEEE Access, 11, 68031-68050.

[48] Fujimoto S, van Hoof H, Meger D. Addressing function approximation error in actor-critic methods. Proceedings of the 35th International Conference on Machine Learning. PMLR; 2019. p. 1587-1596.

[49] Lillicrap, T. P., Hunt, J. J., Pritzel, A., Heess, N. M. O., Erez, T., Tassa, Y., ... & Wierstra, D. P. (2020). U.S. Patent No. 10,776,692. Washington, DC: U.S. Patent and Trademark Office.

[50] Levine S, Kumar A, Tucker G, Fu J. Offline reinforcement learning: Tutorial, review, and perspectives on open problems. arXiv preprint arXiv:2005.01643; 2020.

[51] Moore, D. J. (2025). A Taxonomy of Hierarchical Multi-Agent Systems: Design Patterns, Coordination Mechanisms, and Industrial Applications. arXiv preprint arXiv:2508.12683.

[52] Arora, S., & Doshi, P. (2021). A survey of inverse reinforcement learning: Challenges, methods and progress. Artificial Intelligence, 297, 103500.

[53] Alkhalil, Z., Hewage, C., Nawaf, L., & Khan, I. (2021). Phishing attacks: A recent comprehensive study and a new anatomy. Frontiers in Computer Science, 3, 563060.

[54] Alzaabi, F. R., & Mehmood, A. (2024). A review of recent advances, challenges, and opportunities in malicious insider threat detection using machine learning methods. IEEE Access, 12, 30907-30927.

[55] Abdulkareem, S. A., Foh, C. H., Shojafar, M., Carrez, F., & Moessner, K. (2024). Network intrusion detection: An iot and non iot-related survey. IEEE Access.

[56] Jim, M. M. I. (2024). Cloud Security Posture Management Automating Risk Identification and Response In Cloud Infrastructures. Academic Journal on Science, Technology, Engineering & Mathematics Education, 4(3), 10-69593.

[57] Han X, Pasquier T, Bates A, et al. Unicorn: Runtime provenance-based detector for advanced persistent threats. Network and Distributed System Security Symposium. NDSS; 2020.

[58] Dahl, K. (2024). Asset and Vulnerability Management: Collaboration between and it's uses during conflicts (Master's thesis, NTNU).

[59] Milajerdi SM, Gjomemo R, Eshete B, et al. HOLMES: Real-time APT detection through correlation of suspicious information flows. IEEE Symposium on Security and Privacy. IEEE; 2019. p. 1137-1152.

[60] Baruwal Chhetri, M., Tariq, S., Singh, R., Jalalvand, F., Paris, C., & Nepal, S. (2024). Towards human-ai teaming to mitigate alert fatigue in security operations centres. ACM Transactions on Internet Technology, 24(3), 1-22.