Metrics-Driven Evaluation and Optimization of Honeypots: Toward Standardized Measures of Deception Effectiveness

Zoltán Aradi[1], Sándor Bottyán[2], Eszter Kail[1], Ernő Rigó[1] and Anna Bánáti[1]

[1] John von Neumann Faculty of Informatics, Óbuda University, Bécsi út 96/B, 1034 Budapest, Hungary

aradi.zoltan@uni-obuda.hu; kail.eszter@nik.uni-obuda.hu; rigo.erno@nik.uni-obuda.hu; banati.anna@nik.uni-obuda.hu

[2] Doctoral School of Military Sciences, Ludovika University of Public Service, Hungária krt. 9-11, 1101, Budapest, Hungary

bottyan.sandor@stud.uni-nke.hu

Abstract: Honeypots are widely used to study adversary behavior and support enterprise detection, yet their evaluation is fragmented and often qualitative. This paper proposes a unified, metrics-driven framework assessing honeypots across five dimensions—interaction, data quality, resource use, stealth, and fingerprinting resistance—using hard (observable) and soft (context-dependent) indicators. A normalization and weighting pipeline yields composite scores, while methods combining attack automation, anomaly detection, and ATT&CK enrichment enable reproducible comparisons. Case studies span IT, IoT/OT, and ICS/PLC. Benchmarking guidelines and modern datasets are recommended. An integration roadmap positions honeypot telemetry in SIEM–SOAR–CTI with LLM support and ethical guardrails. Standardized metrics and reproducible methods make honeypot studies comparable, operationally useful and fit for continuous improvement.

Keywords: honeypots; cyber deception; standardized evaluation metrics; benchmarking and reproducibility; fingerprinting resistance; stealth; anomaly detection; SIEM/SOAR integration; cyber threat intelligence (CTI); industrial control systems (ICS); Internet of Things (IoT/IIoT); large language models (LLMs)

1 Introduction

The growth of cyberspace has exposed infrastructures to diverse, sophisticated attacks [1, 5, 22, 46]. Reactive defenses alone cannot counter zero-day exploits, ransomware, botnets, and advanced persistent threats [1, 5, 6, 32].

To address this, researchers employ deception-based defense mechanisms, notably honeypots [2, 3, 5, 42, 61]. Honeypots are decoy systems that mimic real assets to attract and monitor adversaries without jeopardizing production [3, 32, 43, 56, 57, 58]. They enable real-time study of attacks, collection of threat intelligence, and serve as early warnings or distractions [3, 11, 21, 22].

Conventional honeypots face limits: many are finger-printable, and no standardized framework exists for assessing deception or impact [1, 3, 5, 10, 14, 15, 25, 49]. Recent surveys show renewed interest in evaluation [1, 3, 5, 45, 61]. LLM-driven deception and interaction safety further motivate standardized metrics [26, 27, 47, 48, 61]. Advances in cloaking, fingerprinting resistance, protocol fidelity, engagement analysis, ML-based anomaly detection (e.g., Honeyboost) [10, 14, 15, 23], and visualization highlight the need for scalable, standardized methodologies [3, 24, 25].

This paper introduces a unified, metrics-driven framework for evaluating honeypots across five dimensions: interaction, data quality, resource efficiency, stealth, and fingerprinting resistance. It defines measurable indicators and applies them in comparative case studies, providing a reproducible foundation for systematic honeypot assessment. The framework also lays groundwork for adaptive systems using LLMs [26] [27] and provenance-aware audit logging [62][63].

2 Related Work

Honeypots have evolved from simple decoys to advanced deception platforms with integrated analytics [2, 3, 5, 38, 58], reflecting interest in attacker behavior, detection, and optimized deployments [1, 3, 4].

Honeypot Taxonomy. Honeypots are classified by interaction level: low, medium, and high. Low-interaction (LIHs) simulate limited services for tracking automated attacks; medium-interaction (MIHs) simulate OS shells or service responses for deeper intent analysis; high-interaction (HIHs) expose full systems for compromise, yielding rich forensic data but requiring strict containment [3, 37, 56, 57]. Beyond LI/MI/HI, frameworks like T-POT enable modular multi-service deployments [68]; specialized honeypots (e.g., Dionaea, Glastopf) target specific vectors [70] [71]; and HoneyMesh or SpiderTrap extend coverage [5].

Counter-Deception Techniques and Attacker Fingerprinting. Advanced attackers fingerprint honeypots via probing and inconsistencies [10] [61]. Countermeasures include protocol-accurate responses, realistic artifacts, and traffic normalization [2] [15]. Industrial studies show both PLC identification and obfuscation, highlighting an arms race [14] [15]. Adaptive deception that adjusts to attacker behavior is emerging [1] [27].

Evaluation methods lack consensus [1] [3]. Automation frameworks such as HARMer support repeatable tests [25], while suites like T-POT [68] offer rich data but need added methodology for deception-quality assessment. Surveys show many evaluations remain ad hoc, relying on qualitative findings or activity counts instead of reproducible benchmarks, underscoring the need for standardized, multi-dimensional metrics [1] [3]. Fusion and anomaly detection on honeypot logs (e.g., Honeyboost) suggest paths toward metrics-driven pipelines [23].

Table 1
Prior honeypot surveys and taxonomies, showing classification focus and motivating our framework

Year	Authors (Work)	Honeypot Type	Deployment Scale	Key Contributions	Evaluated Metrics	Ref
2004	Provos (Virtual Honeypot Framework)	LI (Honeyd)	Lab	Foundational service emulation	Interaction; protocol coverage	[55]
2012	HoneyMesh (Virtual Honeypot Mesh)	Virtual LI/MI	Design/concept	DDoS mitigation via a mesh of redirecting decoys; early LI/HI hybridization	Redirection success; engagement; overhead	[5]
2016	Zhan et al.	Mixed traces	Research testbed	Statistical characterization of attacks	Rates; session length; distributions	[11]
2020	López–Morales et al.	(HoneyPLC) ICS/PLC	Lab/ICS	Next-gen ICS honeypot design	Protocol fidelity; robustness	[33]
2022	Kandanaarachchi et al. (Honeyboost)	MI logs	Lab dataset	Fusion + anomaly detection over logs	Anomaly rate; temporal divergence	[23]
2022	Ummels et al. (RIoTPot)	Hybrid IoT/OT	Depl./field	Deployable IoT/OT deception	Coverage; realism; footprint	[36]
2023	Ilg et al.	Multi-protocol (T-POT, Dionaea, Cowrie)	Enterprise/large	Survey + comparative analysis	Protocol diversity; logging depth; resources	[3]
2023	Priya & Chakkaravarthy	Containerized/cloud	Cloud cluster	Scalable containerized deception	Scalability; resource usage	[31]
2023	Etcheverry et al.	Identification methods	Mixed	Multistage honeypot identification	Detection accuracy; features	[10]
2024	Zhu et al. (HoneyJudge)	ICS/PLC	Lab/ICS	PLC honeypot identification via memory tests	Detection accuracy; resilience	[14]

2022–2024	T-POT CE (variants; incl. SpiderTrap/ Heralding)	Multi-protocol suite	Enterprise/large	Expanded service set; Elastic-based logging; container orchestration for scale	Protocol diversity; logging depth; storage/CPU	[68]

3 Honeypot Types and Deployment Models

Honeypot classification goes beyond interaction level to include architecture, deployment, and integration with security stacks [1, 3, 5]. This section synthesizes principal types and models, with Table 1 anchoring categories in representative studies and metrics [3, 11, 23].

Software-based honeypots. Software honeypots dominate: LI (Honeyd, Glastopf), MI (Cowrie, Dionaea), and HI (full systems) balance realism, risk, and resource cost [2, 3, 41, 55-57, 69-71]. Cloud-native variants use containers for elasticity but add operational burden [24, 31, 68].

Hardware-based honeypots. In IoT and ICS, physical decoys use device firmware and protocols (e.g., Modbus, BACnet) to capture hardware-specific exploits [4, 16, 72]. They offer high realism but are fragile and costlier than software-only approaches [4] [16]. Large IoT honeypots also reveal brute-force and malware propagation, against embedded devices [4] [8].

Hybrid, virtualized, and distributed deployments. Hybrid designs use low-interaction front ends with selective high-interaction back ends, forwarding traffic for deeper observation [3, 5, 18]. Virtualization via containers/VMs simplifies replication and isolation; T-POT variants broaden protocol coverage and datasets but add overhead [3] [68]. Honeynets link decoys to emulate enterprise/cloud settings, while distributed versions support propagation and coordination studies [3, 11, 18]. Centralized control eases management but risks single points of failure; decentralized designs trade simplicity for resilience [3]. SDN and moving-target defense reconfigure services to increase attacker uncertainty but add complexity [1, 6, 17, 19]. Honeypot telemetry is increasingly integrated with SIEM/SOAR and IDS/IPS for correlation, though governance is needed to avoid analyst fatigue [22, 44, 61, 66, 67]. Large-scale, including hybrid IoT/OT, deployments are feasible but require SOC-grade expertise [3, 22, 36].

4 Defining Measurables: A Metrics Framework

Honeypot assessment needs standardized metrics beyond ad-hoc counts [2, 3, 5]. We propose a five-dimensional framework – interaction, data quality, resource usage, stealth, and fingerprinting resistance – with hard (observable) and soft (context-dependent) indicators [2, 3, 5]. Figures 1-2 show the pipeline and benchmarking; Table 2 lists the metrics.

Metric dimensions and indicators. Interaction level shapes fidelity and risk. Low-interaction systems (e.g., Honeyd) emulate few services [55]; high-interaction (e.g., Cowrie) expose full environments [56] [69]; medium-interaction (e.g., Dionaea) balance safety and observability [3] [70]. Analysts track session length, command diversity, and payload complexity [3, 11, 37]. More interactivity yields richer data but higher risk/cost [3] [56]. Data quality and attribution depend on complete attack sequences, malware diversity, and enrichment with knowledge bases such as MITRE ATT&CK [59]. Fusion and anomaly detection (e.g., Honeyboost) motivate monitoring information content and novelty [23] [40], while realistic artifacts and timing improve engagement [2, 3, 5]. Resource use governs scalability; CPU/RAM, network I/O, and storage footprints are logged for binaries and transcripts. Containerized and flow-based honeypots show typical profiles [24] [31]; stress tests use frameworks like HARMer [25]. Stealth reflects believable interaction, with soft indicators (cloaking success, dwell time) and hard correlates (session duration) [2] [5]. Fingerprinting resistance covers protocol fidelity, realism, and probe failure rates [2, 10, 14, 15]. Adaptive mimicry, randomized timing, and architectural diversity further reduce detectability [3] [31].

Methodological mapping. Hard indicators (session length, command diversity, CPU/RAM, network I/O) are computed with direct measurement and descriptive statistics. Soft indicators (cloaking success, protocol fidelity, realism) use ML-assisted scoring – e.g., anomaly models on temporal features [23] or resilience tests from recent frameworks [10] [14]. Narrative synthesis (ATT&CK mapping, evidence cards) can be LLM-assisted under structured schemas for safety and auditability [26] [27] via protocol fidelity, realism, and probe failure rates [2, 10, 14, 15]. Adaptive mimicry, randomized timing, and architectural diversity further reduce detectability [3] [31].

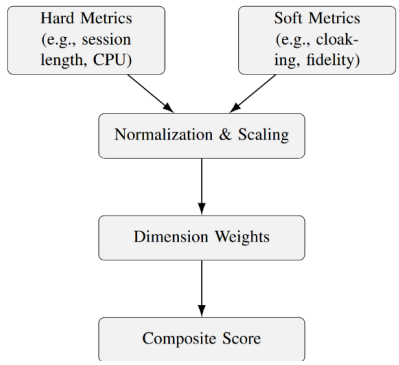

Figure 1
Aggregation pipeline: indicators are normalized, weighted, and combined into a composite effectiveness score used in later benchmarking [3] [5]

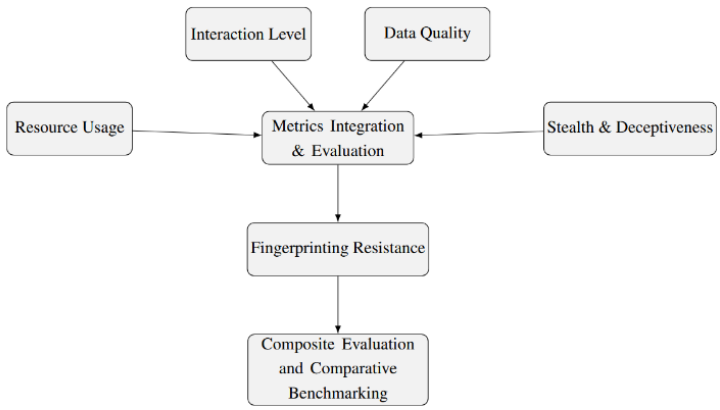

Figure 2
Framework overview: five metric dimensions feed a central engine to produce composite scores for cross-deployment comparison [2, 3, 5]

Table 2
Overview of honeypot evaluation metrics across five dimensions: interaction, data quality, resource usage, stealth, and fingerprinting resistance

	Metric Name	Type	Data Format	Notes
(A) Interaction Level	Session Length	Hard	Time (seconds)	Duration of each attacker session. Longer sessions may indicate greater honeypot realism or deception success.
	Command Diversity	Hard	Count (unique commands)	Behavioral richness of interactions. High values suggest effective engagement.
	Payload Complexity	Hard	Ordinal (Low/Medium/High)	Type and sophistication of binaries or scripts dropped by attackers.

(B) Data Quality & Attack Attribution	Completeness	Soft	Percentage/proportion	Degree to which the full sequence of an attack is captured; higher completeness enables stronger attribution.
	Malicious Payload Collection	Hard	Count (files captured)	Number of malware artifacts or scripts downloaded/executed during sessions.
	Enrichment Capability	Soft	Boolean/scored match	Ability to enrich captured data via threat-intel sources (e.g., VirusTotal, MITRE ATT&CK).
(C) Resource Usage	CPU and RAM Consumption	Hard	Percentage utilization/MB	Processing and memory overhead per honeypot instance.
	Network I/O	Hard	MB/s or packet count	Total incoming/outgoing traffic volumes; helps assess risk and scalability.
	Storage Footprint	Hard	GB/file count	Size and number of logs, binaries, and session captures retained.
(D) Stealth and Deceptiveness	Cloaking Success Rate	Soft	Percentage	Ratio of sessions where the attacker fails to identify the honeypot.
	Attacker Dwell Time	Hard	Time (seconds)	Time an attacker remains active before exit or evasion; proxy for believability.
(E) Fingerprinting Resistance	Protocol Fidelity	Soft	Deviation score/qualitative	Accuracy of protocol behavior emulation (e.g., banners, error codes).
	Environmental Realism	Soft	Qualitative/checklist	Presence of realistic OS artifacts (logs, files, timestamps) that hinder detection.
	Detection Evasion	Hard	Count (failed probes)	Number of attacker fingerprinting attempts that did not identify the honeypot.

5 Evaluation Methodologies

Honeypot assessment must consider interaction depth, stealth, fingerprinting resistance, and intelligence quality, not isolated metrics. Current approaches combine taxonomy-driven models, deception metrics, attack automation, and anomaly detection over multi-source logs [2, 3, 5, 23, 25].

Taxonomy and effectiveness models. These studies frame evaluation around deception fidelity, fingerprinting resistance, and attacker engagement, enabling consistent platform comparison [2] [5]. Open-source surveys benchmark protocol

coverage, logging granularity, and operational utility, mapping to interaction, data quality, and resource usage [3]. Domain-specific work for IoT/IIoT and CPS motivates environment-aware metrics tied to industrial protocols and realism [4, 7, 20]. Concepts from moving-target defense extend to configuration churn and adversary uncertainty [6].

Real-time deception metrics and modeling. Deception quality evolves with attacker behavior and system adaptation [25]. Indicators include behavioral realism, misdirection/confusion rates, and progression depth [2] [5]. Research on PLC obfuscation and identification countermeasures highlights measurable protocol fidelity and probing resilience [14] [15]. Autonomy-oriented work (e.g., reinforcement learning) suggests metrics for state adaptation and reward policies [6] [34]. LLM-driven deception motivates standardized, reproducible evaluation of interactive honeypots [26] [27].

Controlled evaluation via attack automation. HARMer benchmarks protocol realism, fingerprinting resilience, logging accuracy, and resource use through scripted attacks [25].

Time-series anomaly detection. Anomaly detection quantifies deviations from learned behavior. Honeyboost fuses heterogeneous logs and applies time-series models (including autoencoders) to detect novel tactics or strategy shifts [23]. Metrics include temporal divergence, behavioral variance, and engagement irregularities, strengthening data quality and stealth. Statistical methods further characterize attack distributions and uncertainty [11].

Metric collection and analysis in case studies. Case studies extract metrics from session transcripts and telemetry from Cowrie, Dionaea, and T-POT [41, 68-70], enriched with packet captures and flows via Zeek and Wireshark [73] [74]. Host-level monitors log CPU, memory, and I/O for resource usage, while protocol/behavioral artifacts inform stealth and fingerprinting resistance [10] [14]. These multi-source datasets enable consistent computation of hard (e.g., session length, command counts) and soft (e.g., deception success, protocol fidelity) indicators, aggregated and scored within the framework (Figures 1-2, Table 2).

6 Case Studies and Comparative Analysis

Five representative deployments demonstrate the framework, applying Table 2 metrics and the aggregation pipeline in Figure 1. The cases cover SSH cloaking, multi-protocol engagement, cloud-native orchestration with ML analytics, a hybrid IoT/OT setting (RIoTPot), and an ICS/PLC honeypot [3, 4, 16, 23, 31]. Judgments are based on normalized indicators and interpreted with Figure 2.

SSH honeypot fingerprinting resistance. Cowrie is a widely used SSH/Telnet honeypot evaluated by command diversity, session duration, and logging depth [3].

Identification studies show unrealistic artifacts, timing, or banners enable fingerprinting; cloaking with realistic environments and timing normalization improves engagement and reduces abandonment [10]. These adjustments enhance interaction, data quality, and stealth in the framework [2, 3, 10].

T-POT, Cowrie, and Dionaea span multiple protocols, balancing fidelity, logging depth, and overhead [3, 68, 69, 70]. Dionaea captures diverse binaries but is fingerprintable; session and command distributions aid comparison [3, 10, 11].

Cloud-based honeypots offer elastic scaling and isolation but consume more resources [31]. Fusion-based anomaly detection (e.g., Honeyboost) learns baselines and surfaces deviations, improving data quality and stealth [23]. Tamper-evident logging further supports trustworthy forensics [60].

Industrial IoT honeypots. In ICS/IIoT, protocol-aware deception (e.g., Modbus/BACnet) and PLC honeypots face targeted identification; studies show both obfuscation and active tests [14, 15, 33]. Surveys note brute-force and probing of embedded devices, stressing realistic telemetry for completeness but with higher cost and fragility [4] [16]. Such settings score high in interaction and data quality, but lower in scalability and resources.

Comparative summary. Table 3 synthesizes the five deployments on a qualitative Low/Medium/High scale derived from normalized measurements (e.g., session duration, commands captured, fingerprint-probe failure rate, resource utilization). Rows correspond to the framework's metric categories, columns to deployment types, and the entries reflect the composite interpretation outlined in Figures 1 and 2 [3, 23, 31].

These cases show how the framework enables evaluation across diverse deployments but also highlight a key limitation: the lack of standardized, reproducible benchmarks [3] [5]. This motivates the following discussion on benchmarking and validation.

Table 3

Comparison of representative deployments using the proposed metrics (Table 2); values are qualitative summaries from normalized indicators (Figures 1-2)

Metric	**SSH Cloaking**	**Multi-Protocol (T-POT)**	**Cloud + ML**	**RIoTPot (IoT/OT)**	**ICS/PLC**
Interaction Level	Medium	Medium	Mixed	Medium	High
Data Completeness	High	Med.–High	High	Med.–High	High
Cmd./Behavior Diversity	High	Medium	High	Medium	Medium
Resource Usage	Low–Med.	Medium	High	Medium	High
Stealth	High	Variable	High	Medium	Variable
Fingerprint Resistance	Strong	Mixed	Adaptive	Mixed	Medium

7 Role of Honeypots in CTI, SIEM, and SOAR Architectures

Honeypots generate telemetry (e.g., IPs, flows, IoCs) that, once normalized and correlated, can be elevated into operational and strategic intelligence [21] [22]. In the framework, these streams inform data quality, stealth, and fingerprinting resistance, and should be exchanged using open standards for interoperability [64] [65].

CTI role and outputs. In CTI, honeypots act as passive sensors emitting observables and IoCs, shareable via STIX/TAXII [64] [65]. SOC and CTI studies show such feeds aid in characterizing TTPs and producing reports and risk assessments [21] [22]. Figure 3 illustrates the path from honeypot data to CTI enrichment and SOC/CSIRT action [21, 22, 64, 65].

SIEM/SOAR ingestion and automation. SIEMs aggregate logs and serve as the main ingestion point for honeypot telemetry [22]. JSON/CEF/LEEF simplify parsing, and tools like Wazuh or TheHive integrate with STIX/TAXII [64, 65, 66, 67]. SOAR systems orchestrate enrichment, triage, and containment (e.g., EDR queries, lookups, ticketing), reducing MTTR and workload [22]. Fusion and anomaly detection over honeypot logs (e.g., Honeyboost) improve signal quality [23, 40], while automation frameworks (e.g., HARMer) support repeatable playbook testing [25]. These roles align with Figure 3 and Table 2.

Operational challenges. High-interaction honeypots can generate large event volumes that strain SIEM storage and risk alert fatigue without tuning [22]. False positives and inconsistent normalization slow triage, while over-filtering may drop rare but useful signals, motivating careful parser design and retention policies [22] [23]. Distributed deployments also require secure transport, identity management, and reproducible playbook testing [22] [25].

Performance indicators inside SIEM–SOAR. To gauge honeypot impact in SIEM–SOAR (per Table 2), we focus on four outcomes: MTTD/MTTR reduction from honeypot-driven detections and automation [22]; data-volume efficiency, i.e., ratio of high-value to total events [22]; novel IoC yield promoted to CTI via STIX/TAXII [21, 64, 65]; and false-positive reduction from fusion and anomaly scoring (e.g., Honeyboost) [23]. These map to the composite-scoring approach and interfaces in Figure 3.

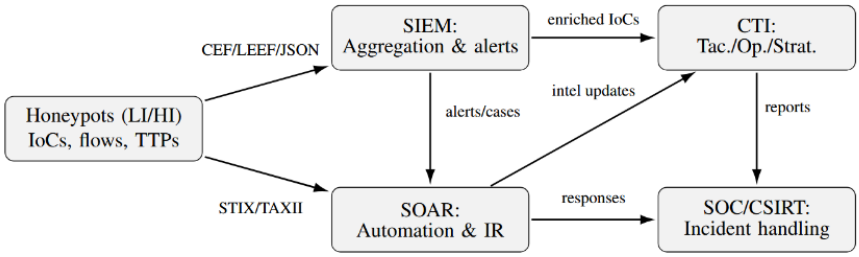

Figure 3
End-to-end interfaces among honeypots, SIEM, SOAR, and CTI. Normalized honeypot events (CEF/LEEF/JSON) and standardized threat data (STIX/TAXII) flow through correlation and orchestration to enrich CTI and guide SOC/CSIRT response [21, 22, 64, 65, 66, 67].

Summary. Integrated with SIEM and SOAR, honeypots expand detection, shorten response, and enrich CTI with novel indicators, if normalization and automation manage volume while preserving signal quality [21-23, 25, 64-67]. Figure 3 shows the interfaces.

8 Discussion

This section summarizes benchmarking guidance, integration plans, ethical guardrails, and takeaways to improve transparency and reproducibility. It links back to Table 2, Figures 1-2, and situates telemetry in enterprise workflows (Figure 3).

Benchmarking and validation. This topic remains difficult due to heterogeneous deployments and uneven scoring. We recommend versioned configs, public scripts, and recent datasets (e.g., CICIoT2023, TON_IoT, Bot-IoT, CSE-CIC-IDS2018, LITNET-2020) [50-54], with uncertainty reporting and cost-sensitive SOC metrics.

System integration and near-term roadmap. Future work should align honeypot telemetry with SIEM–SOAR–CTI pipelines [21-23, 25-27, 60, 64, 65]. Ingestion normalizes events, enriches with STIX/TAXII IoCs, and maps to ATT&CK TTPs (Table 2) [21, 22, 59, 64, 65]. A decision engine correlates signals by timing, rarity, reputation, and metric scores, producing normalized bundles (Figures 1-2) [11, 23, 28, 29, 30]. The LLM layer uses these bundles to draft summaries, hypotheses, playbooks, and notes under safe-evaluation guidance [26] [27]. Analyst interfaces post results to IR/ticketing (e.g., TheHive/ServiceNow), while SOAR playbooks run in guarded modes [22, 60, 67]. These stages ensure the LLM augments, not replaces, deterministic gates [21, 22, 60].

Data, governance, and evaluation rigor. Training data includes honeypot logs, SIEM events, tickets, sandbox detonations, curated negatives, and synthetic traffic [23] [25]. Labels cover operational outcomes (true positive, benign, duplicate, informational, noise), with weak labels from IoC confidence and TTP alignments plus a human-reviewed subset [21, 22, 64, 65]. Features combine temporal,

behavioral, context, and deception signals; cross-network transfer builds on recent embedding work [28]. Controls include deduplication, PII scrubbing, drift monitoring, and data sheets aligned with systems-engineering guidance [21, 22, 60]. Evaluation applies temporal splits, cost-sensitive metrics (e.g., precision@k), and shadow-mode SOC trials before automation [22, 26, 28].

Safety guardrails and SOC operations. To cut false positives and analyst load, promotions require multi-signal consensus (honeypot + reputation/EDR); thresholds scale with risk; calibrated scores allow abstention; canaries catch unsafe automation; and active learning targets uncertain cases [21-23, 25, 60]. LLMs serve as Tier 0.5 assistants drafting ATT&CK summaries and playbooks, with Tier 1 approval and Tier 2/3 handling containment [22] [59]. Effectiveness is tracked via MTTD/MTTR, analyst time, duplicate suppression, and enrichment rates; error budgets trigger advisory-only fallback [22]. Immutable audit trails log prompts, model versions, and decisions [21] [60]. In ICS/OT, automation defaults to observe-and-advise and stays confined to decoys, per deception and fingerprinting work [12, 14, 16, 60]. Milestones: M1 – data path & shadow mode; M2 – ranking + LLM summarizer; M3 – guarded automation with drift calibration; M4 – A/B workload studies [22, 25, 26].

Standardization, trade-offs, and limits. No platform-agnostic evaluation method exists; definitions of engagement quality, stealth, and dwell time remain fragmented, requiring reproducible baselines and uncertainty reporting [1-5, 11]. High-interaction designs provide rich data but expand attack surface and risk fingerprinting; PLC studies show small inconsistencies can expose decoys [9, 13, 14, 16]. Low-interaction lowers risk/cost but misses advanced tactics [3, 16]. Cloud/container deployments add elasticity but need careful tuning [24, 31, 68]. The framework aligns stealth, scalability, and data quality with objectives, but faces unvalidated weightings (Figure 1), soft-indicator bias, and uncertain long-term stability [22, 64, 65].

ML, anomaly detection, and LLM use. Data–fusion and anomaly–detection over honeypot logs (e.g., Honeyboost) learn temporal baselines and can surface novel tactics in near real time [23]. Classical learners and transfer methods support correlation/prioritization [28-30], and studies of learning–based alert triage show operational gains when embedded in SOC workflows [34] [35]. LLMs help with semantic summarization, TTP mapping, and controlled decoy interaction when strictly bound to structured inputs and auditable prompts [26, 27, 39]; risks (bias, misclassification, adversarial evasion) and integration cost require calibration, abstention policies, and clearly bounded action scopes [22] [60].

Ethical and legal considerations. Ethical/legal guidance includes lawful basis, data minimization, PII scrubbing, safe malware handling, and restricting deception to decoys. Publish configs and logs responsibly, with IRB/ethics approvals and tamper-evident audits [21, 60].

Table 4
Possible LLM integration points across the honeypot–SIEM–SOAR–CTI pipeline. Roles are paired with structured inputs and guardrails to ensure reproducibility and safety. Citations point to surveys, frameworks, and empirical systems that motivate each integration.

Pipeline stage	LLM role (examples)	Structured inputs & safety guardrails	Primary refs.
Ingestion / Normalization	Schema validation helper; IoC extraction from logs; ATT&CK mapping notes	JSON/CEF/LEEF events; STIX/TAXII indicators; deterministic parsers precede LLM; human-in-the-loop on schema drift	[21, 22, 59, 64, 65]
Correlation / Prioritization	Natural-language feature synthesis; hypothesis generation; similarity search across past cases	Scored bundles from decision engine; LLM suggestions remain advisory while ranking uses classical ML (e.g., XGBoost, Random Forests)	[11, 22, 23, 29, 30]
Anomaly Triage (Honeyboost-like)	Explain anomalies; propose follow-up queries; translate temporal divergences into analyst notes	Time-series anomaly scores from fusion models; no raw PCAPs—bounded schema only	[22, 23]
Interactive Deception / LLM Honeypots	Policy-constrained dialog in decoy services; believable banners/responses; red-team scripting support	Prompt whitelisting; rate limits; output filters; execution in sandboxed/decoy networks to contain risk	[5, 26, 27]
SOAR Playbooks & IR Support	Playbook selection and parameter filling; action justification; case summarization	Case context from SIEM; guardrails per systems-security engineering with mandatory human approval for risky actions	[22, 60]
CTI Production & Reporting	Narrative synthesis; TTP mapping; drafting STIX bundles for sharing	Validated indicators and evidence cards; analyst review before publication to CTI stores	[21, 22, 59, 64, 65]
Evaluation & Benchmarking Workflow	Auto-generate test scenarios; A/B summaries; dataset documentation (“data sheets”)	HARMer-generated traffic; shadow-mode trials; precision@k and workload metrics; reproducible audit trails	[25, 26, 28]

Takeaways. The proposed metrics (Table 2) and aggregation logic (Figures 1-2) enable reproducible assessment across SSH honeypots, multi-protocol suites, cloud deployments, and ICS/IIoT, moving beyond ad-hoc counts [1-2, 3, 5, 8, 11, 16, 22, 24, 31]. Integrated with SIEM–SOAR–CTI (Figure 3), standardized indicators and

open formats support defensible investment, sharing, and analytics [21, 22, 64, 65]. Near-term priorities include coupling scores with correlation models and guarded LLM use; longer-term work requires validation, cost-sensitive weighting, and mixed synthetic/real evaluation (e.g., HARMer) to quantify MTTD/MTTR and workload [25-30].

Conclusions

This work proposed a unified, metrics-driven framework for honeypot evaluation, structured around interaction, data quality, resource efficiency, stealth and fingerprinting resistance. Case studies across SSH, multi-protocol, cloud-native, IoT/OT, and ICS/PLC environments confirmed its applicability and revealed trade–offs between realism, scalability, and detectability. The framework moves beyond ad-hoc or qualitative assessments by providing reproducible, comparable indicators. It also outlines how honeypot telemetry can be integrated with SIEM, SOAR, and CTI pipelines to improve detection, response, and threat intelligence. The proposed methodology thus, meets its objective of standardizing honeypot evaluation and supporting operational adoption.

Future work will focus on refining metric weighting, validating across larger datasets, and incorporating adaptive deception with anomaly detection and LLM support.

References

[1] P. Beltrán-López, M. Gil Pérez, and P. Nespoli. Cyber deception: taxonomy, state of the art, frameworks, trends, and open challenges. IEEE Communications Surveys & Tutorials, August 2025

[2] A. Javadpour, F. Ja'fari, T. Taleb, M. Shojafar, and C. Benzaïd. A comprehensive survey on cyber deception techniques to improve honeypot performance. Computers & Security, 140:103792, 2024

[3] N. Ilg, C. R. Koch, L. Schreyer, C. Kreibich, A. Klenze, M. Zohner, and F. Kargl. A survey of contemporary open-source honeypots, frameworks, and tools. Journal of Network and Computer Applications, 220:103939, 2023

[4] J. Franco, A. Aris, B. Canberk, and A. S. Uluagac. A survey of honeypots and honeynets for Internet of Things, Industrial Internet of Things, and Cyber–Physical Systems. IEEE Communications Surveys & Tutorials, 2021

[5] R. Skopik, G. Settanni, and R. Fiedler. Three decades of deception techniques in active cyber defense—retrospect and outlook. Computers & Security, 106:102288, 2021

[6] J.-H. Cho, D. P. Sharma, H. Alavizadeh, S. Yoon, C. Alcaraz, and T. H. Kim. Toward proactive, adaptive defense: a survey on moving target defense. IEEE Communications Surveys & Tutorials, 22(1):709-745, 2020

[7] T. Humayed, J. Lin, F. Li, and B. Luo. Cyber–physical systems security—a survey. IEEE Internet of Things Journal, 4(6):1802-1831, 2017

[8] S. Torabi, Đ. Klisura, J. Khoury, E. Bou–Harb, C. Assi, and M. Debbabi. Internet–wide analysis, characterization, and family attribution of IoT malware: a comprehensive longitudinal study. IEEE Transactions on Dependable and Secure Computing, 22(2):1703-1716, 2025

[9] E. Passino and F. Mantziou. Nested Dirichlet models for unsupervised attack pattern detection in honeypot data. Annals of Applied Statistics, 19(1):586-613, 2025

[10] E. Etcheverry, V. Busino, G. Calabria, and L. De Giusti. A multistage framework for honeypot identification. Digital Threats: Research and Practice, 5(4):42:1-42:28, 2023

[11] Z. Zhan, M. Xu, and S. Xu. Characterizing honeypot-captured cyber attacks: statistical framework and case study. IEEE Transactions on Information Forensics and Security, 11(11):2365-2379, 2016

[12] D. J. S. Raja, N. Hemavathi, R. Sriranjani, and P. Arulmozhi. Integrated game-theoretic and honeypot-based distributed denial of service attack detection and mitigation in advanced metering infrastructure. IEEE Transactions on Instrumentation and Measurement, 74:5503710, 2025

[13] P. S. Mrudula, R. D. A. Raj, A. Pallakonda, Y. R. M. Reddy, K. K. Prakasha, and V. Anandkumar. Smart grid intrusion detection for IEC 60870-5-104 with feature optimization, privacy protection, and honeypot-firewall integration. IEEE Access, 13:128938-128958, 2025

[14] H. Zhu, M. Liu, B. Chen, X. Che, P. Cheng, and R. Deng. HoneyJudge: a PLC honeypot identification framework based on device memory testing. IEEE Transactions on Information Forensics and Security, 19:6028-6043, 2024

[15] S. Maesschalck, W. Fantom, V. Giotsas, and N. Race. These are not the PLCs you are looking for: obfuscating PLCs to mimic honeypots. IEEE Transactions on Network and Service Management, 21(3):3623-3635, 2024

[16] R. Bridges, J. Hopley, and S. Bhatia. Don’t get stung, cover your ICS in honey: how do honeypots fit within industrial control system security. Computers & Security, 114:102598, 2022

[17] M. F. Saiyed and I. Al-Anbagi. A game theoretic model for strategic defence selection against DDoS attacks in IoT networks. IEEE Transactions on Network and Service Management, July 2025

[18] L. Nguemkam, A. H. Anwar, V. K. Tchendji, D. K. Tosh, and C. Kamhoua. Optimal honeypot allocation using core attack graph in partially observable stochastic games. IEEE Access, 12:187444-187455, 2024

[19] K. Horák, B. Bošanský, P. Tomášek, C. Kiekintveld, and C. Kamhoua. Optimizing honeypot strategies against dynamic lateral movement using

partially observable stochastic games. Computers & Security, 87:101579, 2019

[20] Y. Li, Y. Xiao, Y. Li, and J. Wu. Which targets to protect in critical infrastructures—a solution from network science perspective. IEEE Access, 6:56214-56221, 2018

[21] A. Spyros, I. Koritsas, A. Papoutsis, P. Panagiotou, D. Chatzakou, and D. Kavallieros. AI-based holistic framework for cyber threat intelligence management. IEEE Access, 13:20820-20846, 2025

[22] C. Vielberth, A. Böhm, V. Fichtinger, and G. Pernul. Security operations center: a systematic study and open challenges. IEEE Access, 8:227756-227779, 2020

[23] S. Kandanaarachchi, H. Ochiai, and A. Rao. Honeyboost: boosting honeypot performance with data fusion and anomaly detection. Expert Systems with Applications, 201:117073, 2022

[24] S. C. Sethuraman, T. G. Jadapalli, D. P. V. Sudhakaran, and S. P. Mohanty. Flow-based containerized honeypot approach for network traffic analysis: an empirical study. Computer Science Review, 50:100600, 2023

[25] E. Huang, C. Korbely, J. R. Crandall, Y. Shoshitaishvili, Z. Zhao, S. A. Hofmeyr, and A. Doupé. HARMer: cyber attacks automation and evaluation framework. IEEE Access, 8:129397-129414, 2020

[26] S. B. Weber, M. Feger, and M. Pilgermann. Don't stop believin': a unified evaluation approach for LLM honeypots. IEEE Access, 12:144579-144587, 2024

[27] A. Kouremetis, A. Ioannidis, G. Loukas, C. Xenakis, and G. Theodorakopoulos. MIRAGE: cyber deception against autonomous cyber attacks. Annals of Telecommunications, pp. 1-15, 2024

[28] L. Gioacchini, M. Mellia, L. Vassio, I. Drago, G. Milan, and Z. Ben Houidi. Cross-network embeddings transfer for traffic analysis. IEEE Transactions on Network and Service Management, 21(3):2686-2699, 2024

[29] T. Chen and C. Guestrin. XGBoost: a scalable tree boosting system. In Proc. 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '16), pp. 785-794, 2016

[30] L. Breiman. Random forests. Machine Learning, 45(1):5-32, 2001

[31] V. S. D. Priya and S. S. Chakkaravarthy. Containerized cloud-based honeypot deception for tracking attackers. Scientific Reports, 13:1437, 2023

[32] M. Nawrocki, A. King, T. C. Schmidt, M. Wählisch, and J. Schönfelder. SoK: a data-driven view on methods to detect reflective amplification DDoS using honeypots. In Proc. IEEE European Symposium on Security and Privacy (EuroS&P), 2023

[33] E. López-Morales, C. Rubio-Medrano, A. Doupé, Y. Shoshitaishvili, R. Wang, T. Bao, and G.-J. Ahn. HoneyPLC: a next-generation honeypot for industrial control systems. In Proc. 2020 ACM SIGSAC Conference on Computer and Communications Security (CCS '20), pages 279-291, November 2020

[34] L. Tong, A. Laszka, C. Yan, N. Zhang, and Y. Vorobeychik. Finding needles in a moving haystack: prioritizing alerts with adversarial reinforcement learning. In Proc. AAAI Conference on Artificial Intelligence, 34(1):946-953, 2020

[35] S. McElwee, J. Heaton, T. Fraley, and J. Cannady. Deep learning for prioritizing and responding to intrusion detection alerts. In Proc. 2017 IEEE Military Communications Conference (MILCOM), 2017

[36] R. Ummels, R. van Rijswijk-Deij, R. Struik, and S. Hesselman. RIoTPot: a hybrid IoT/OT network traffic deception model for deployable honeypots. In Proc. 38th Annual Computer Security Applications Conference (ACSAC), pp. 766-781, 2022

[37] F. Shamsi, A. Khajeh, M. Conti, R. Perdisci, M. Antonakakis, and M. Polychronakis. Measuring and clustering network attackers using medium-interaction honeypots. In Proc. IEEE European Symposium on Security and Privacy Workshops (EuroS&PW), pp. 294-306, 2022

[38] W. Fan, Z. Du, D. Fernández, and V. A. Villagra. Enabling an anatomic view to investigate honeypot systems: a survey. IEEE Systems Journal, 12(4):3906-3919, 2018

[39] W. Z. A. Zakaria and L. M. Kiah. A review of dynamic and intelligent honeypots. ScienceAsia, 39S:001-010, 2013

[40] A. Abdou, J. Szefer, E. Shi, and R. Karri. HoneyModels: machine learning honeypots. In Proc. MILCOM 2021 – IEEE Military Communications Conference, pp. 886-891, November 2021

[41] M. Valicek, O. Krammer, R. Rusnak, and T. Hubinsky. Creation and integration of remote high interaction honeypots. In Proc. 2017 International Conference on Software Security and Assurance (ICSSA), pp. 50-55, July 2017

[42] M. S. Rana and M. A. Shah. Honeypots in digital economy: an analysis of intrusion detection and prevention. In Proc. IET Conference Proceedings, pp. 91-98, October 2021

[43] F. Zhang, M. Zulkernine, and A. Haque. Honeypot: a supplemented active defense system for network security. In Proc. 8th International Scientific and Practical Conference of Students, Post-graduates and Young Scientists (MTT'2002), 2002

[44] V. Mahajan and S. K. Peddoju. Integration of network intrusion detection systems and honeypot networks for cloud security. In Proc. 2017 International Conference on Computing, Communication and Automation (ICCCA), pp. 829-834, May 2017

[45] K. E. Silaen, R. Rahim, and A. P. Wibawa. Usefulness of honeypots towards data security: a systematic literature review. In Proc. 2023 International Workshop on Artificial Intelligence and Image Processing (IWAIIP), pp. 422-427, December 2023

[46] M. Čerger and J. Hudec. Cyber-security threats origins and their analysis. Acta Polytechnica Hungarica, 10(9), 2023

[47] Á. Balogh, M. Érsok, E. Kail, and A. Bánáti. Possibilities for optimization of real-time attacker profiling with honeypots. In Proc. 2025 IEEE 19th International Symposium on Applied Computational Intelligence and Informatics (SACI 2025), Timişoara, Romania, pp. 1-6, May 2025

[48] M. Érsok, Á. Balogh, E. Kail, and A. Bánáti. Adaptive deception architectures: conceptual foundations for LLM-powered honeypot systems. In Proc. 2025 IEEE 19th International Symposium on Applied Computational Intelligence and Informatics (SACI 2025), Timişoara, Romania, pp. 1-6, May 2025

[49] P. Patel, A. Dalvi, and I. Siddavatam. Exploiting honeypot for cryptojacking: the other side of the story of honeypot deployment. In Proc. 2022 6th International Conference on Computing, Communication, Control and Automation (ICCUBEA), Pune, India, August 2022

[50] E. C. P. Neto, A. H. Lashkari, et al. CICIoT2023: a realistic IoT intrusion detection dataset and benchmark. Sensors, 23(13):5941, 2023

[51] N. Moustafa, M. Keshk, E. Debie, and H. Janicke. Federated TON_IoT Windows datasets for evaluating AI-based security applications. arXiv preprint arXiv:2010.08522, 2020

[52] N. Koroniotis, N. Moustafa, E. Sitnikova, and B. Turnbull. Towards the development of realistic botnet dataset in the IoT for network forensic analytics: Bot-IoT. arXiv preprint arXiv:1811.00701, 2019

[53] M. Sharafaldin, A. H. Lashkari, and A. A. Ghorbani. CSE-CIC-IDS2018 on AWS: a realistic intrusion detection dataset. Canadian Institute for Cybersecurity (UNB), 2018

[54] LITNET-2020: a real-world network flow dataset for network intrusion detection. Electronics, 9(5):800, 2020

[55] N. Provos. A virtual honeypot framework. In Proc. 13th USENIX Security Symposium, San Diego, CA, USA, pp. 1-14, 2004

[56] N. Provos and T. Holz. Virtual Honeypots: From Botnet Tracking to Intrusion Detection. Addison-Wesley, Boston, MA, USA, 2007

[57] L. Spitzner. Honeypots: Tracking Hackers. Addison-Wesley, Boston, MA, USA, 2003

[58] The Honeynet Project. Know Your Enemy: Learning About Security Threats, 2nd edition. Addison-Wesley, Boston, MA, USA, 2004

[59] D. A. Strom, F. A. Lacey, M. H. Padilla, et al. MITRE ATT&CK: design and philosophy. MITRE Technical Report, 2018

[60] R. Ross, M. McEvilley, and J. C. Oren. Developing cyber-resilient systems: a systems security engineering approach. NIST Special Publication 800-160, Vol. 2, Rev. 1, 2021

[61] S. E. Heckman and F. J. Stech (eds.) Cyber Denial, Deception and Counter-Deception: A Framework for Supporting Active Defense. Springer, Cham, Switzerland, 2015

[62] S. Haber and W. S. Stornetta. How to time-stamp a digital document. Journal of Cryptology, 3(2):99-111, 1991

[63] B. Schneier and J. Kelsey. Secure audit logs to support computer forensics. ACM Transactions on Information and System Security, 2(2):159-176, 1999

[64] OASIS. STIX Version 2.1. OASIS Standard, 2021

[65] OASIS. TAXII Version 2.1. OASIS Standard, 2021

[66] Wazuh. Wazuh Security Platform, 2024

[67] TheHive Project. TheHive: Security Incident Response Platform, 2024

[68] Telekom Security. T-POT: the all-in-one honeypot platform, 2024

[69] Cowrie Developers. Cowrie SSH and Telnet Honeypot, 2024

[70] DinoTools. Dionaea: a framework for collecting malware samples, 2024

[71] Mushorg. Glastopf: web application honeypot, 2024

[72] Mushorg. Conpot – ICS/SCADA honeypot, 2024

[73] The Zeek Project. Zeek network security monitor, 2024

[74] Wireshark Foundation. Wireshark network protocol analyzer, 2024