Phase-adaptive Model Routing in LLM-driven SSH Honeypots: A Systematic Mapping Review

Raiymbek Magazov [1,2], Fatima Uralova [1,*], Kuanysh Abeshev [3], Bagdat Yagaliyeva [1], Zhuldyz Alimseitova [1], and Yernar Shamuratov [1]

[1] Department of Cybersecurity, Institute of Automation and Information Technology, Satbayev University, Almaty, Kazakhstan
[2] Department of Cybersecurity and Cryptology, Faculty of Information Technology and Artificial Intelligence, Al-Farabi Kazakh National University, Almaty, Kazakhstan
[3] School of Digital Technologies, Almaty Management University, Almaty, Kazakhstan
Email: magazovraiko@gmail.com (R.M.); f.uralova@satbayev.university (F.U.); k.abeshev@almau.edu.kz (K.A.); b.yagaliyeva@satbayev.university (B.Y.); zh.alimseitova@satbayev.university (Zh.A.); samuratovernar77@gmail.com (Y.Sh.)
*Corresponding author

***Abstract*—Large Language Model (LLM)-driven Secure Shell (SSH) honeypots have rapidly displaced static, templated decoys, yet they share a structural limitation: a single model and a single policy serve every turn of an intrusion, from low-stakes reconnaissance to high-value persistence. This review asks whether honeypots can instead adapt their inference to the attacker's phase. We conduct a Preferred Reporting Items for Systematic Reviews and Meta-Analyses (PRISMA) 2020-guided systematic mapping review of fifty influential studies (2020–2026) drawn from three strands—honeypots and generative deception, cost-aware model routing and mixture-of-experts inference, and LLM agents, embeddings, and retrieval-augmented generation in security operations—assembled under a disclosed, purposive top-50 prioritization ranked by age-normalized citation velocity. That prioritization is a deliberate mapping strategy: it surfaces the field's most influential work rather than claiming exhaustive coverage, and a sensitivity analysis tests how robust the conclusions are to the cap. Three findings emerge. Across the reviewed literature, honeypot adaptivity has advanced at the policy and generation layers but not at the inference-allocation layer; the routing literature has solved capacity allocation but conditions only on query-intrinsic signals, not on adversarial session phase; and embedding-based analytics, corroborated by a production deployment, show the phase signal is observable and reliable. The strands are individually mature and mutually disconnected. We formalize this gap and propose a phase-adaptive, agent-oriented framework that pairs an embedding-based phase estimator and an O(1) phase-to-model dispatch layer with an unprivileged, campaign-level observation substrate. The review contributes a reproducible corpus, an age-normalized bibliometric ranking, a formal gap analysis, and a conceptual architecture that operationalizes the maturation arc from static to phase-adaptive honeypots.**

***Keywords*—Secure Shell (SSH) honeypot, large language model, cyber deception, model routing, attack-phase adaptation, mixture-of-experts, command embeddings, systematic mapping review**

I. Introduction

Honeypots remain a primary instrument for observing intrusion behavior, but the fidelity of what they capture is bounded by how convincingly they respond. For two decades that response was templated and static; an attacker who probed beyond the script saw the seams. The arrival of Large Language Models (LLMs) changed the ceiling: a model can now synthesize plausible shell output on demand, and a wave of generative honeypots has followed. The published anchor for this review, Magazov *et al.* [1] pairs an LLM Secure Shell (SSH) honeypot with embedding-based command-stage analytics over a six-month, four-million-event deployment, reporting an F1 score (the harmonic mean of precision and recall) of 0.90 for stage recognition—evidence that an attacker's phase is both real and machine-recognizable.

A limitation cuts across this new generation of systems. They apply one model and one policy uniformly across every phase of an intrusion. Reconnaissance, which is high-volume and low-stakes, is served by the same expensive inference as exploitation or persistence, which are rare and high-value. A recent state-of-knowledge synthesis [2] makes the point explicitly: adaptive inference is absent from current LLM honeypots. Meanwhile a parallel literature on cost-aware model routing and Mixture-of-Experts (MoE) inference has solved exactly this allocation problem—but only on query-intrinsic signals such as difficulty or length, not on the temporal structure of an attack.

The cost of ignoring phase is not hypothetical. Honeypot traffic is dominated by automated reconnaissance and brute-force activity—high-volume,

Manuscript received June 30, 2026; revised July 23, 2026; accepted August 25, 2026; published October 22, 2026.

low-information turns that a small model can answer convincingly—while the rare turns of exploitation and persistence are exactly where response fidelity decides whether an adversary stays engaged. Serving both with one model forces a single operating point onto a workload with a bimodal value distribution: either capacity is wasted on reconnaissance or fidelity is starved on the turns that matter. A honeypot able to estimate phase and route inference accordingly would resolve this tension, and it would do so using a signal the deception setting uniquely supplies—the attacker's own command sequence, observed in order.

This review connects the two. We systematically survey fifty influential studies and argue that the field is converging on a third stage in a clear maturation arc: static honeypots, then uniform single-model LLM honeypots, then phase-adaptive, agent-oriented honeypots.

The review is deliberately cross-disciplinary. Prior surveys treat LLM honeypots, model routing, and security-oriented LLM analytics as separate literatures; we read them together because the gap is precisely a gap between them. Existing systematizations of the honeypot field are valuable but scoped differently from this review: recent surveys and a Systematization of Knowledge catalogue LLM-honeypot designs at the systems and architecture level—interaction paradigms, protocol coverage, prompting and deployment strategies [2–4]—yet they treat the language model as a single fixed component and do not examine how inference capacity is allocated across an intrusion.

Our contribution is orthogonal to theirs: rather than re-survey system architectures, we isolate the inference-allocation layer that those architecture-level surveys leave uniform, which is precisely where the gap this review formalizes resides. We do not attempt an exhaustive census of any single strand. Instead we assemble the most influential work across all three, measured on one consistent scale, and ask what an architecture sited at their intersection would require: (1) A PRISMA 2020–reported systematic mapping review of fifty influential works spanning honeypots, model routing, and security-oriented LLM analytics, with a transparent, purposive selection protocol and a sensitivity analysis of its influence-ranked cap. (2) An age-normalized citation-velocity ranking with a dual established/recent cohort, so works from 2020 and 2026 are comparable. (3) A formal gap analysis showing inference-allocation adaptivity is unaddressed. (4) A phase-adaptive, agent-oriented conceptual framework that closes the gap.

The research questions of this paper are: RQ1: How have SSH/LLM honeypot architectures evolved over 2020–2026? RQ2: What model-routing, cascading, and MoE techniques exist, and on what signals do they condition? RQ3: To what extent do current LLM honeypots adapt inference to the attacker's phase? RQ4: What architecture integrates phase-adaptive routing into an agent-oriented honeypot?

The rest of the paper is structured as follows: Section II sets out the primitives. Section III details the review methodology. Section IV reports the bibliometric and thematic findings. Section V consolidates the cross-study extraction synthesis and formalizes the gap. Section VI presents the proposed framework. Section VII discusses open challenges and limitations, and Section VIII provides the conclusion.

II. Background

A. SSH Honeypots and Deception

Honeypots are classified by interaction depth. Low- and medium-interaction systems emulate a constrained service; high-interaction systems expose a fuller environment at greater risk. Cowrie remains the de facto SSH/Telnet baseline, and large-scale deployments continue to use it to quantify brute-force and scripted traffic with low-latency templated responses [5]. The realism ceiling of templated emulation is the standing motivation for generative approaches.

Two consequences follow from interaction depth. Higher interaction yields richer telemetry but enlarges the attack surface the defender must contain, which is why production deployments lean on low- and medium-interaction systems despite their lower fidelity. And because attackers actively fingerprint decoys—probing for the timing regularities, missing utilities, and canned errors that betray emulation—per-response realism is not cosmetic; it sets how long an adversary stays observable. Generative honeypots raise that ceiling, but, as Section IV shows, they raise it uniformly rather than where engagement risk is highest.

B. Model Routing, Cascading, and Mixture-of-Experts

When several models of differing cost and capability are available, an inference system can route each request to the cheapest adequate model (routing), escalate only hard requests (cascading), or activate a sparse subset of experts per token (MoE). FrugalGPT [6] and RouteLLM [7] are canonical exemplars. The shared aim is to preserve quality while reducing cost—the same economic logic a honeypot faces when it decides how much model capacity an attacker's turn deserves.

The three mechanisms differ in where the decision is made. A router decides before generation, dispatching the whole request to one model; a cascade decides during, escalating when a cheaper model's own answer is judged inadequate; an MoE gate decides within the model, activating a sparse subset of experts per token. All three reduce to a learned mapping from an observed feature to a capacity choice, and all are evaluated on a cost-quality frontier. The honeypot setting adds a feature none of them currently uses—a position in an adversarial interaction—so the dispatch primitive a honeypot needs is a small, fast map from estimated phase to model tier, closer to a router's pre-generation decision than to token-level gating.

C. Attack Phases and the Kill Chain

Intrusions progress through recognizable phases—reconnaissance, exploitation, persistence—that map onto kill-chain and MITRE Adversarial Tactics, Techniques, and Common Knowledge (ATT&CK) abstractions. Surveys of autonomous, LLM-driven attacks [8] formalize

this progression for the agentic threat model. Phase is the signal this review argues honeypots should, but do not, condition their inference on.

Treating phase as a latent variable recoverable from observed commands is the conceptual move the framework builds on. Reconnaissance commands enumerate the environment, exploitation commands attempt privilege or code execution, and persistence commands install footholds; each leaves a distinctive lexical and structural signature in the command stream. The embedding and intrusion-detection work of Section IV-D shows such signatures are recoverable with standard encoders, and the production deployment of Magazov *et al.* [1] shows the resulting phase estimate is accurate enough to act on. Phase is thus both operationally meaningful and statistically observable—the two properties an adaptive controller requires of its input.

Four terms recur throughout and are used in a precise, non-interchangeable sense. An adaptive honeypot changes its behavior in response to the attacker along any axis, whether policy, response content or engagement. A phase-adaptive honeypot adapts specifically to the estimated attack phase and, in this work, does so at the inference-allocation layer by routing to a model tier conditioned on phase. Agent-oriented denotes a honeypot designed to observe and characterize autonomous, LLM-driven attackers rather than only human operators. Campaign-level denotes taking the multi-session campaign, rather than the single session, as the unit of observation and analysis; campaign-level is the canonical form used throughout, and the variants campaign-oriented and campaign-aware that appear elsewhere in the literature are not used here. The four terms name distinct and separately verifiable properties and are never used interchangeably: adaptive is always the generic superordinate and never an abbreviation of phase-adaptive; phase-adaptive always denotes adaptation at the inference-allocation layer; agent-oriented always denotes the class of adversary a honeypot is built to observe; and campaign-level always denotes the unit of analysis. A system may hold any combination of the four. The framework of Section VI is simultaneously phase-adaptive in its routing layer and agent-oriented and campaign-level in its observation substrate, whereas the reinforcement-learning honeypots of Section IV-B are adaptive without being phase-adaptive—the distinction on which the gap analysis of Section V turns.

III. Review Methodology

We conduct this study as a systematic mapping review: its aim is to chart how three mature but disconnected literatures relate to one another and to locate the structural gap between them, rather than to enumerate exhaustively every study in any single strand. We adopt PRISMA 2020 reporting for the screening funnel, and we test the robustness of our conclusions to the purposive cap through a sensitivity analysis (Section III-E). The protocol below is reported so the corpus can be regenerated deterministically from the search record.

A. Search Strategy

We searched Scopus, IEEE Xplore, and Web of Science on 23 June 2026, combining honeypot and deception terms with LLM, model-routing, and attack-phase terms, restricted to 2020–2026. Database hits totaled 1,055, with 18 records added by snowballing.

The query was built around three term groups joined by conjunction: a deception group (honeypot, deception, decoy, Cowrie), a model group (large language model, LLM, model routing, cascade, mixture-of-experts), and a behavior group (attack phase, kill chain, reconnaissance, persistence, command). The three databases were chosen for complementary coverage—Scopus and Web of Science for journal and indexed-conference breadth, IEEE Xplore for the security and networking venues where honeypot work concentrates. Because the field moves largely through preprints, we treated arXiv records surfaced by snowballing as eligible provided they met the same screening criteria, and we recorded the version of record wherever one had since appeared.

B. Eligibility and the Purposive Cap

Studies were included if they addressed LLM-driven or adaptive honeypots, model routing/MoE inference, or security-oriented LLM analytics relevant to attack-phase reasoning. We applied a deliberate, disclosed purposive cap: rather than attempting a census, we retain the fifty most influential works, with influence measured by age-normalized citation velocity (Section III-D). The cap is a transparency-controlled design choice, not a coverage claim; the threshold is applied in the eligibility stage and is reported as such. Framing the work as a mapping review, rather than as a review claiming exhaustive coverage, states plainly what the citation-velocity prioritization does and does not claim.

The cap serves two ends. It keeps the synthesis tractable and current in a literature that is both large and fast-moving, and it forces an explicit, reproducible influence criterion rather than an implicit one. We state its cost plainly: any influence-based cap under-represents very recent work that has not yet accrued citations and niche work that is important but lightly cited. We mitigate the first with the dual-cohort design of Section III-D, which ranks recent and established work separately, and we accept the second as the price of a transparent, bounded corpus.

Exclusion reasons were recorded per record so the funnel in Fig. 1 is auditable. We further note that citation velocity is not, in practice, a binding exclusion filter: the included corpus itself contains seven 2026 studies with zero recorded citations, admitted on relevance. The purposive cap governs breadth rather than imposing a citation floor, which is the premise the sensitivity analysis of Section III-E tests directly.

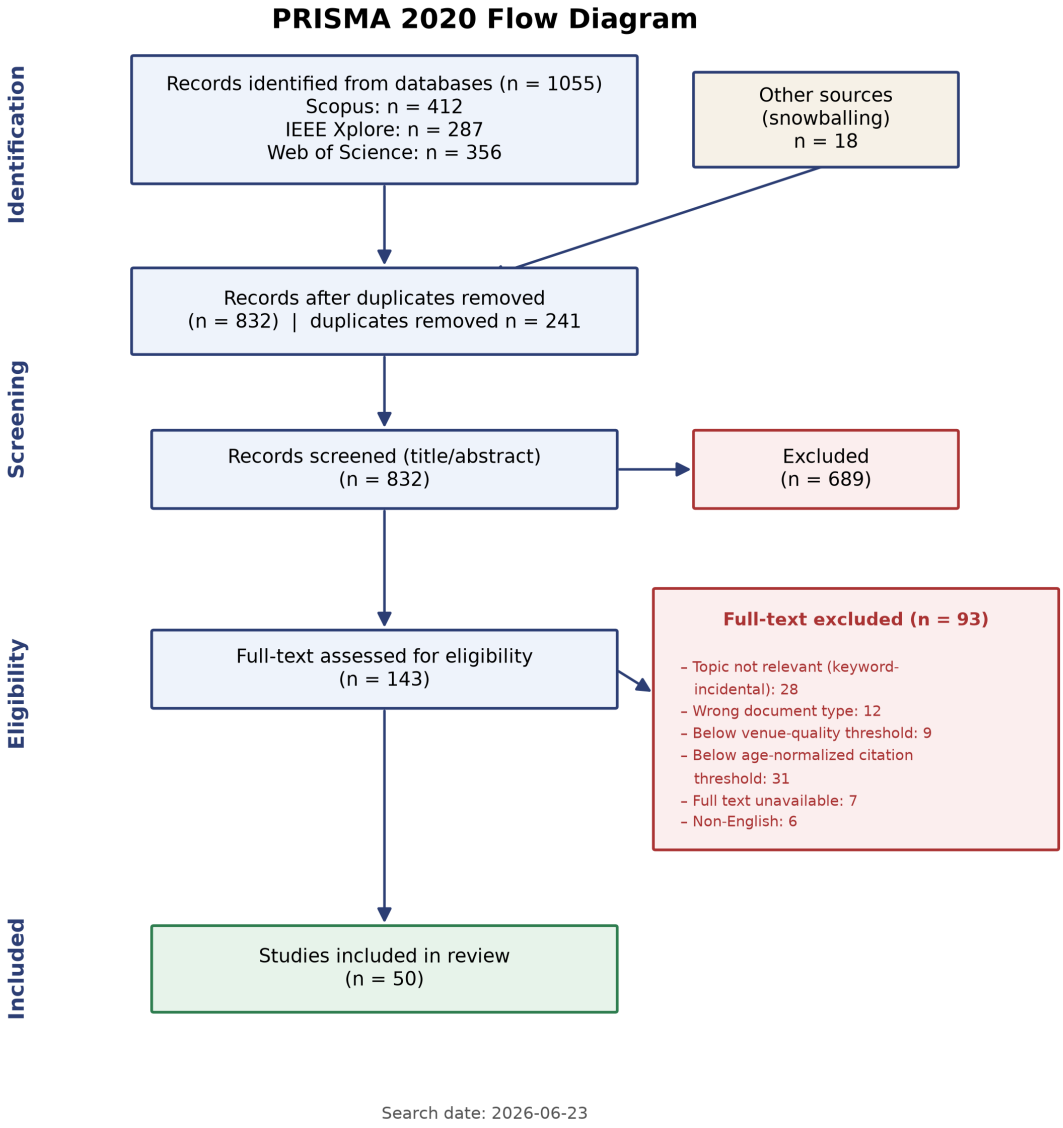


Fig. 1. PRISMA 2020 flow diagram. The funnel reduces 1,073 identified records to the 50 included studies; the age-normalized citation-velocity threshold operationalizes the disclosed purposive top-50 cap.

C. *Selection*

After removing 241 duplicates, 832 records were screened on title and abstract, of which 689 were excluded. The remaining 143 were assessed in full text; 93 were excluded (topic-incidental, wrong document type, below venue-quality threshold, below the age-normalized citation threshold that operationalizes the purposive cap, full text unavailable, or non-English), leaving 50 included studies. Fig. 1 shows the funnel.

D. *Extraction and Ranking*

Each study was extracted into a structured schema (Table I): bibliographic fields, honeypot type, LLM backend, reported latency and fidelity metrics, core contribution, controlled keywords, and an index-verified citation count. Bibliographic metadata and citation counts were resolved against bibliographic indexes and never hand-entered; unresolved fields were flagged rather than imputed. Because raw citation counts are not comparable across a 2020–2026 window, we rank by citation velocity (citations per year) and report a dual cohort—established (2020–2023) versus recent (2024–2026).

TABLE I. PER-STUDY DATA-EXTRACTION SCHEMA

Field	Description
id	Stable per-study reference number
authors / year / title	Bibliographic identity
venue / DOI	Publication venue and index-resolved identifier
honeypot_type	Interaction class / paradigm (e.g. SSH LLM; static; RL)
llm_backend	Model(s) used; “none” for classical systems
latency_ms	Reported response latency, where stated
fidelity_metrics	Realism / engagement metrics reported
core_contribution	One-line statement of contribution
citations / cohort	Index-verified count; established vs. recent
keywords	Controlled terms for co-occurrence analysis

Note: Fields of the per-study data-extraction schema applied to all 50 included studies. Source: review protocol of Section III-D; the table is definitional and reports no study-level data. Statistical basis: not a statistical summary. Abbreviations: DOI, Digital Object Identifier; LLM, large language model; RL, reinforcement learning; SSH, Secure Shell.

Velocity is computed as citation count divided by the number of years since publication, floored at one year so 2026 entries are not divided by zero. This is a deliberately simple normalization; its only purpose is to stop a 2020 paper's accumulated citations from dominating a 2025 paper of comparable per-year impact. The integrity protocol is strict: every bibliographic field and citation count was resolved against a bibliographic index, version-of-record DOIs were substituted where the index carried them, and any field the index could not return was left blank and flagged rather than estimated. No citation, DOI, or metric in the corpus was hand-entered or inferred.

E. Sensitivity Analysis of the Purposive Cap

Because the central finding of this review is the absence of a coupling between honeypot adaptivity and phase-conditioned inference routing, a purposive cap raises a specific risk: the coupling might exist in a lower-velocity study that the cap excluded, making the gap an artifact of selection rather than a property of the field. We test this directly. Relaxing the cap entirely, we issued an unrestricted search over OpenAlex (2020–2026) using the three strand queries of Section III-A together with a set of targeted bridge queries—for example, "honeypot model routing", "honeypot phase-adaptive inference", and "adaptive inference honeypot attack stage"—constructed specifically to surface any honeypot-plus-routing coupling at any citation level, with no velocity threshold applied.

After deduplication against the fifty included studies, this returned 1,255 additional eligible works. Of these, 655 fall below the lowest citation velocity in the ranked corpus (10.0 citations per year), 131 below 1.0 citations per year and 48 carry no recorded citations at all, so the expanded pool is weighted precisely toward the low-visibility work an influence-ranked cap is accused of suppressing.

Screening this expanded pool for the coupling itself, the bridge queries flagged 26 candidates. Exactly one pairs honeypot deception with phase-conditioned, model-tier routing: Magazov *et al.*'s companion study [9], which is excluded from the review corpus by design (Section VI) and carries zero citations at the search date—precisely the kind of low-velocity item the cap would drop. No independently authored study, at any citation level, bridges the two strands. Most apparent honeypot-and-"routing" matches resolve on inspection to network-layer routing, in ad-hoc, MAC-layer and black-hole-attack studies, rather than to routing of language-model inference.

The screen did, however, surface a cluster of genuinely relevant recent work that the cap had excluded, and we incorporate it rather than set it aside. ShellGames [10] attacks the same latency ceiling directly, pairing speculative command execution with the routing of complex commands to a sandboxed execution environment; critically, it routes to an execution backend rather than across model tiers, and it does not estimate the attacker's phase. Two further 2026 preprints—one rethinking cyber deception for AI attackers, one applying reinforcement learning and language models to deception in contested network environments—target autonomous adversaries but likewise leave inference allocation uniform, as do the earlier adaptive-deception orchestrators built on reinforcement learning. These works narrow the gap this review identifies; none closes it. The central conclusion is therefore robust to the cap, and we report the near-misses explicitly so that the reader can judge that robustness rather than take it on trust.

F. Study-Quality Appraisal

To characterize the evidential strength of the corpus rather than only its influence, we appraised every included study against a transparent four-axis rubric, each axis scored 0–2 for a total of 0–8. The axes are: evidentiary rigour, meaning the presence of quantitative evaluation metrics and comparative or statistical validation; data scale and realism, meaning whether evaluation rests on a real-world or large-scale deployment, a bounded benchmark, or neither; peer-review status, distinguishing a peer-reviewed venue from an indexed preprint or an unindexed record; and reproducibility and transparency, meaning whether a named dataset or benchmark and, ideally, released artifacts are reported.

Because a full re-reading of fifty papers lies outside the scope of a mapping review, the rubric is applied deterministically to characteristics extracted from each study's abstract and metadata, and the scoring rules are published with the corpus so that the appraisal is reproducible and adjustable. This is a lightweight, characteristic-based appraisal rather than a full CASP or AMSTAR instrument, and the transparency axis is bounded by what abstracts disclose about artifact availability.

The corpus is of moderate-to-high evidential strength. The mean total score is 4.8 of 8 (median 5), with 21 studies in the high band (6–8), 22 moderate (3–5) and 7 low (0–2). Axis means are 1.1 for rigour, 1.2 for scale, 1.5 for peer-review status and 1.0 for transparency, and 27 of the 50 studies appear in peer-reviewed venues. Per-study scores accompany the corpus as supplementary material. The appraisal matters for the review's central claim: the gap is not an artifact of a few weak studies but is visible across a body of predominantly peer-reviewed, empirically evaluated work.

IV. Findings

A. Corpus Overview and Bibliometrics

The fifty studies span 2021–2026, with 48 of 50 (96%) published from 2022 onward and a pronounced concentration in 2024–2025. The dual-cohort split is deliberately skewed recent—6 established (2020–2023) against 44 recent (2024–2026)—reflecting how abruptly the LLM turn reshaped the field rather than any sampling artifact. Provenance is mixed, and we distinguish two senses of it. Twenty-nine of the fifty were hosted on arXiv at the search date; by venue of record, however, 27 studies are peer-reviewed (ACL Anthology, IEEE, Elsevier, MDPI) and 23 remain preprints, which is the split used in the quality appraisal of Section III-F. Three carry no DOI because their venues had not yet registered one. We report this distribution because it conditions how citation evidence should be read—preprints accrue citations on a

different clock than archival venues, which is precisely why we rank by velocity rather than raw counts.

Fig. 2 shows the year-by-type trend, and it is the empirical spine of the maturation arc. Static and classical honeypots stay flat across the window; the uniform single-model LLM honeypot category rises sharply, to 9 studies in 2024 and 11 in 2025; and phase-adaptive or agent-oriented designs barely register, appearing only once in 2024 and twice in each of 2025 and 2026. That third category groups two distinct properties because each is individually too rare to plot on its own; the grouping is a coding convention of the trend analysis and not a claim that phase-adaptive and agent-oriented are the same property, which Section II-C keeps apart. The shape is unambiguous: the field has industrialized the generative honeypot but has not yet moved to phase-conditioned inference.

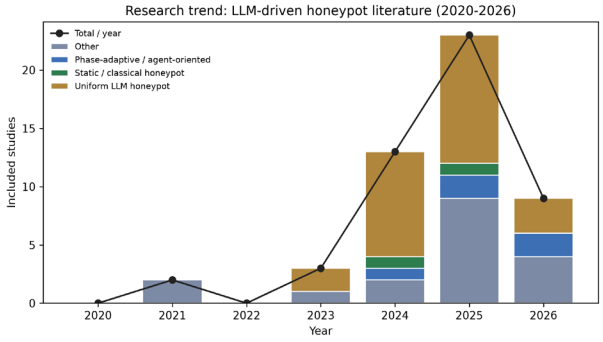


Fig. 2. Publication trend by honeypot category, 2020–2026. Uniform single-model LLM honeypots rise to 9 (2024) and 11 (2025), while designs that are phase-adaptive or agent-oriented—grouped as a single coding category, see Section II-C—remain marginal and classical honeypots stay flat.

By citation velocity (Table II), whose Strand column marks each entry as honeypot-specific or as general-purpose routing, MoE, security-LLM or survey work, the leaders are routing and efficiency works rather than honeypots: RouteLLM at 224 citations/year, FrugalGPT at 223.7, quality-aware Hybrid-LLM routing at 163.5, and the OLMoE mixture-of-experts model at 133. The highest honeypot-specific entry, shelLM, trails at roughly 21 citations/year, and the published SSH-honeypot anchor is too recent to have accrued comparable counts. Read carefully, this gap is not a weakness of the deception literature but a signal: the capacity-allocation machinery this review proposes to import is mature, intensively used, and validated at scale—just not yet inside a honeypot.

The keyword co-occurrence structure (Fig. 3) corroborates the three-strand separation. Honeypot, ssh, and deception co-occur densely (honeypot–deception = 10, honeypot–ssh = 7); the routing terms form a second dense cluster (model routing–cost = 4, model routing–llm = 9); and a third cluster links intrusion detection, semantic, and embeddings (semantic–embeddings = 4, embeddings–intrusion detection = 4). The honeypot and model-routing clusters are effectively disjoint across the fifty studies. We are careful about how much this shows: because the search strategy deliberately treated the three strands as distinct term groups (Section III-A), a degree of non-co-occurrence is expected by construction, so we read the co-occurrence structure as corroborating the gap rather than as independent statistical proof of it.

The load-bearing evidence is the content-level absence documented in the sensitivity analysis (Section III-E) and the extraction synthesis of Section V-A, which Section V-B formalizes.

TABLE II. MOST INFLUENTIAL INCLUDED STUDIES, RANKED BY AGE-NORMALIZED CITATION VELOCITY (SEARCH YEAR 2026)

Rank	Study	Year	Cites/yr	Strand
1	Ong *et al.* [7]	2024	224.00	Routing
2	Chen *et al.* [6]	2023	223.67	Routing
3	Ding *et al.* [11]	2024	163.50	Routing
4	Muennighoff *et al.* [12]	2024	133.00	MoE
5	Javadpour *et al.* [3]	2024	69.00	Survey (deception)
6	Jitkrittum *et al.* [13]	2025	66.00	Routing
7	Zhang *et al.* [14]	2025	49.00	Routing
8	Wang *et al.* [15]	2025	48.00	Routing
9	Xu *et al.* [8]	2025	28.00	Survey (security-LLM)
10	Koukoulis *et al.* [16]	2025	23.00	Security-LLM
11	Sladic *et al.* [17]	2023	21.00	Honeypot
12	Mohammadshahi *et al.* [18]	2024	16.50	Routing
13	Otal *et al.* [19]	2024	15.50	Honeypot
14	Shirkavand *et al.* [20]	2025	13.00	Routing
15	Lekssays *et al.* [21]	2025	10.00	Security-LLM

Note: The 15 highest-ranked of the 50 included studies. Citation counts were resolved against Scopus, IEEE Xplore and Web of Science on 23 June 2026, with no value hand-entered or imputed (Section III-D). Citation velocity is the index-verified count divided by years since publication, the denominator floored at one year; Strand is the controlled category assigned during extraction (Table I). Abbreviations: Cites/yr, citations per year.

B. *Honeypots: From Static to Adaptive to LLM-driven*

Classical SSH/Telnet honeypots remain the deployment baseline: Cowrie-based studies continue to quantify large-scale brute-force and scripted traffic with low-latency templated responses [5], and their realism ceiling is well documented. Two adaptivity programs pushed past static templates without generative language models. Reinforcement-learning approaches optimize engagement and resource use—HoneyIoT learns interaction policies for high-interaction Internet of Things (IoT) decoys [22], Q-Cowrie casts SSH engagement as a Q-learning/MDP problem [23], and deep-RL designs harden honeypots against runtime denial-of-service [24]; game-theoretic optimal-stopping formalizes when to divert or terminate an intruder [25]. Surveys of cyber-deception techniques [3] and a PRISMA review of AI-enhanced honeypots [4] confirm the trajectory while noting that adaptivity is typically expressed as policy selection over fixed response generators, not as adaptation of the generative fidelity itself.

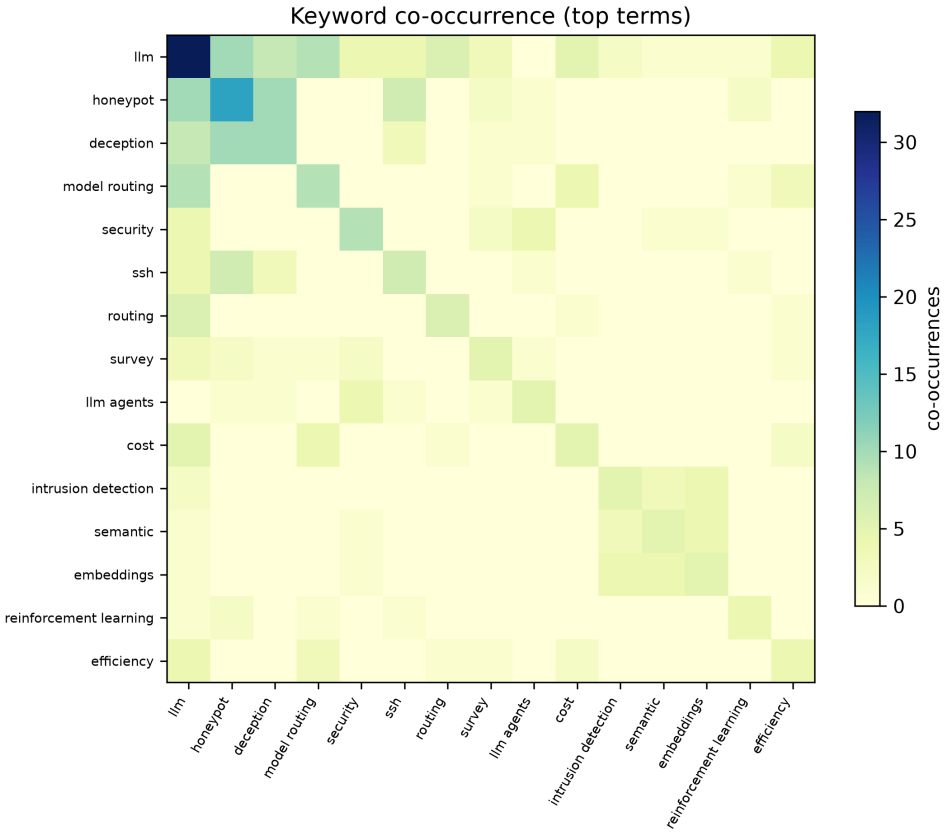


Fig. 3. Controlled-keyword co-occurrence across the 50 included studies. The honeypot/ssh/deception cluster and the model-routing/cost cluster are dense internally but effectively disjoint from each other.

The dominant recent development is the generative honeypot, in which a language model synthesizes shell output. shelLM established session-stateful generative SSH/Linux shells with high expert-rated plausibility [17]; subsequent systems fine-tuned open models for interactive SSH/Telnet [19], profiled the latency–fidelity trade-off across thirteen backends (LLMHoney) [26], extended the paradigm to industrial control system (ICS) and programmable logic controller (PLC) protocols (LLMPot) [27], terminals (HoneyGPT) [28], multi-protocol frameworks (VelLMes) [29], Lightweight Directory Access Protocol (LDAP) [30], and general shells (HoneyLLM) [31]; beyond the host, deception is now orchestrated at the infrastructure layer, as Koney does for Kubernetes clusters [32]. The published anchor for this review, Magazov *et al.* [1], advances the line by pairing an LLM SSH honeypot with embedding-based command-stage analytics, reporting F1 = 0.90 stage recognition over a six-month, four-million-event deployment.

A recent Systematization of Knowledge (SoK) [2] synthesizes the area and observes explicitly that adaptive inference is absent. The LLM Agent Honeypot [33] foreshadows the next step by instrumenting decoys to capture autonomous AI attackers. Most recently ShellGames [10], surfaced by the sensitivity analysis of Section III-E rather than by the influence-ranked cap, attacks the latency ceiling directly: it pairs speculative command execution with the routing of complex commands to a sandboxed execution environment. It routes to an execution backend rather than across model tiers and does not estimate the attacker's phase, so it narrows the gap identified here without closing it.

Read across the extraction (Table AI in Appendix), the generative-honeypot subgroup is methodologically uneven in exactly the dimensions that matter for phase adaptation. Reported fidelity is a patchwork: shelLM reports roughly 92% deception accuracy with expert plausibility judgments on GPT-3.5/4 [17]; LLMHoney profiles exact-match, Bilingual Evaluation Understudy (BLEU), and cosine similarity against latency across thirteen backends spanning Gemma, Phi, and Gemini [26]; the published anchor reports F1 = 0.90 stage recognition from embedding-based command analytics [1]; and several systems report only qualitative realism or engagement length on backends ranging from fine-tuned Llama-3 [19] to hosted commercial APIs [28–31].

Two absences are conspicuous. None of the generative systems reports fidelity broken out by attack phase, and only LLMHoney foregrounds the latency cost that a router exists to manage. The classical and RL-driven systems, by contrast, optimize a single scalar—engagement or capture rate—and hold the response generator fixed [5, 22–24]. The subgroup thus proves that high-fidelity generation is achievable while leaving the allocation of that fidelity across an intrusion entirely unaddressed.

C. Cost-Aware Model Routing, Cascading, and MoE

A parallel and, to date, disconnected literature solves exactly the allocation problem honeypots ignore. Cascades

trade cost for quality by escalating only hard queries (FrugalGPT) [6], and learned routers dispatch each query to the cheapest adequate model: RouteLLM from preference data [7], quality-aware Hybrid-LLM routing [11], universal [13], dynamic mixed-pool [15], RL-based multi-round [14], cost-aware contrastive [20], and cost-effective routing across diverse models (Routoo) [18]. MoE systems internalize routing as sparse expert gating—OLMoE [12], layerwise-recurrent [34] and threshold [35] routers, low-rank expert ensembles [36], and LLM-as-router gating [37]—while a recent survey [38] and the HyDRA heterogeneous-pool architecture [39] consolidate the design space. Crucially, every routing signal here is query-intrinsic (difficulty, length, embedding similarity); none conditions on an evolving adversarial session phase.

The extraction makes the conditioning signal explicit for each system. Preference-trained routers [7], quality-aware binary routers [11], contrastive [20], universal [13], and dynamic mixed-pool [15] routers all condition on properties of the incoming query—predicted difficulty, expected answer quality, or embedding similarity to a calibration set. Cascades condition on the confidence of a cheaper model's own answer [6]; MoE gates condition on per-token hidden states [12, 34, 35, 37]; multi-round routers add a reinforcement-learned controller over the same query-local features [14].

Across all fifteen routing and MoE entries the conditioning variable is internal to the request, and not one conditions on where in an interaction sequence the request falls. The reported gains—cost reductions of large magnitude at preserved quality, with the velocity leaders in Table II drawn entirely from this group—are exactly the gains a phase-conditioned honeypot would seek. What the routing literature lacks is the temporal, adversarial signal that a honeypot, uniquely, has in abundance.

D. LLM Agents, Threat Intelligence, and Embedding-Based Analytics in Security Operations

Beyond generative deception and capacity allocation, a third body of work repositions the large language model from a producer of attacker-facing content to an instrument of defensive analysis. This strand is methodologically adjacent to our proposal: it establishes that security-relevant command and event semantics are machine-recoverable, machine-labelable, and operationally actionable—precisely the properties on which an attack-phase estimator depends.

1) LLMs for threat intelligence and detection engineering

A first cluster automates the Cyber-Threat-Intelligence (CTI) and detection-engineering pipeline. Mitra *et al.* [40] introduce FALCON, an autonomous agent that mines unstructured CTI and compiles it into deployable intrusion-detection rules, demonstrating that the extraction-to-signature workflow can be driven end to end by an LLM. Complementarily, Bertiger *et al.* [41] subject LLM-generated detection rules to systematic evaluation, quantifying the gap between syntactically valid and operationally precise signatures and thereby tempering claims of fully autonomous rule synthesis. Alhuzali [42] reframes the same task as an investigative one, grounding threat-intelligence reasoning in retrieved evidence to improve the faithfulness of generated analyses.

These advances are counterbalanced by Meng *et al.* [43], who expose concrete vulnerabilities in LLM-assisted CTI, showing that adversarially crafted inputs can degrade or manipulate downstream intelligence. For the present framework, this cluster confirms that LLMs can reliably parse and annotate security-relevant text, while Meng *et al.* [43] supplies a standing caution: any phase estimator or CTI dependency must itself be treated as an adversarially reachable component.

2) Retrieval-augmented generation in cybersecurity

A second cluster addresses the grounding problem directly through retrieval-augmented generation (RAG), which conditions model output on curated security knowledge. Liu and Anwar [44] embed retrieval within a multi-agent incident-response loop (AutoBnB-RAG), coordinating specialized agents over a shared evidence store, while Lekssays *et al.* [21] propose TechniqueRAG, aligning free-text threat reports with MITRE ATT&CK techniques via retrieval and learned reranking—a capability that maps directly onto the ATT&CK-labeled telemetry our observation substrate emits. The reliability and safety of such pipelines remain open problems: Barnes [45] measures the calibration of incident-response agents under adversarial evidence, surfacing systematic overconfidence, and Xu *et al.* [46] provide a structured taxonomy of attacks against, and defenses for, RAG systems. Taken together, Liu and Anwar [44], Lekssays *et al.* [21] establish that ATT&CK-grounded retrieval is sufficiently mature to instrument a honeypot, whereas Barnes [45], Xu *et al.* [46] delimit the trust boundary within which such instrumentation must operate.

3) Transformer- and embedding-based attack analysis and intrusion detection

A third cluster provides the representational foundation for phase estimation. Chhetri and Namin [47] employ transformer models to predict the downstream consequences of cyber attacks, evidencing that attack semantics are encoded in learnable sequence representations. At the detection layer, Koukoulis *et al.* [16] apply self-supervised contrastive transformer learning to intrusion detection, and Mahdaouy *et al.* [48] contextualize NetFlow-based detection with deep representations, both recovering discriminative structure from raw traffic without dense labeling. Othman *et al.* [49] demonstrate that sentence-transformer embeddings can align natural-language attack descriptions with structured vulnerability references, while Liu and Guo [50] couple semantic analysis with self-supervised embeddings for IoT and sensor-network intrusion detection.

Finally, Biswas *et al.* [51] attach attribution to encoder-based LLM detectors, rendering their decisions explainable. Collectively, Refs. [16, 47–51] substantiate the three properties our estimator presupposes: command- and event-level text yields discriminative embeddings,

those embeddings align with controlled security vocabularies, and the alignment is stable and attributable enough to drive a routing decision. The phase estimator of Section VI is thus a direct transposition of methods already validated on adjacent security tasks, rather than an untested construction.

V. Synthesis and Research Gap

This section formalizes the research gap, in two steps. Section V-A consolidates the per-study extraction quantitatively and establishes what the fifty included studies do and do not measure; Section V-B states the gap those measurement absences imply. The extraction synthesis is presented here, rather than as a further findings subsection, so that the empirical evidence sits directly alongside the claim it supports and is not stated twice.

A. Cross-Study Extraction Synthesis

Table AI (Appendix) consolidates the per-study extraction across all fifty works—honeypot type, LLM backend, reported latency, fidelity metrics, and core contribution. To move beyond narrative reading, Table III aggregates the extraction fields quantitatively across all fifty studies; the counts, rather than any single exemplar, surface three structural patterns, each of which becomes a design requirement in Section VI.

Two entries in Table III are themselves findings rather than description. Numeric response latency is reported by exactly one of the fifty studies, so a latency-by-honeypot-type comparison is not computable from the literature; rather than impute values we report that sparsity, because it is direct quantitative evidence for the measurement gap this review identifies. Fidelity resolved by attack phase is reported by none of the fifty. The honeypot and routing subgroups also fail to share a comparison axis—the former reports realism and engagement, the latter cost–quality trade-offs—which is why a phase-weighted fidelity metric has to be introduced rather than borrowed.

TABLE III. Quantitative Synthesis of the Extraction Fields (N = 50)

Extraction field	Distribution across the fifty included studies
Strand composition	Honeypot and deception systems 16; model routing, MoE and security-oriented LLM analytics 28; surveys and SoK 6
Honeypot subgroup by backend	LLM-based generative 9; non-generative (static, RL, orchestration) 7
Numeric response latency reported	1 of 50
Fidelity resolved by attack phase	0 of 50
Fidelity metric families in the honeypot subgroup (a study may report more than one)	Expert-rated realism or plausibility 7; engagement or capture statistics 6; task-specific reward or mitigation rate 3; exact-match, BLEU and cosine similarity 1; stage-level F1 1

Note: Distribution of the extraction fields across all 50 included studies, aggregated by the authors from the per-study extraction of Table AI (Appendix). Entries are counts of studies, not counts of reported measurements. The strand and honeypot-subgroup rows are mutually exclusive and exhaustive within their scope; the fidelity-metric row is not, because one study may report several metric families, so its entries sum to more than the 16 studies of the honeypot subgroup. Abbreviations: N, number of included studies.

First, reported fidelity is heterogeneous and almost never phase-resolved. Deception accuracy, BLEU and cosine similarity, expert plausibility, and engagement length each appear, but they are reported as session- or corpus-level aggregates; not one study in the corpus reports fidelity separately for reconnaissance, exploitation, and persistence. A system cannot be optimized along an axis it never measures, which is why the framework adopts phase-weighted fidelity as a first-class metric.

Second, latency is reported sporadically and is essentially never traded off against phase. Only the real-time SSH systems treat response latency as a headline quantity [26], even though latency is the exact cost term the entire routing strand of Section IV-C exists to manage. The corpus therefore contains, in separate communities, both the cost metric and the machinery to optimize it, but not together in any honeypot in the reviewed corpus.

Third, the unit of analysis is overwhelmingly the session, not the campaign. Few studies track an adversary across kill-chain transitions, and per-command MITRE ATT&CK labeling is confined to the embedding/CTI strand [21] rather than the honeypots themselves. This limits visibility into exactly the phase dynamics a phase-adaptive system must observe to act. These three absences—phase-blind fidelity, decoupled latency, and session-bounded observation—are the measurement surface the proposed framework targets directly.

B. From Measurement Absences to the Research Gap

Read against that measurement surface, three findings emerge. First, honeypot adaptivity has advanced at the policy layer (RL, game theory) and the generation layer (LLMs) but not at the inference-allocation layer—no system in the reviewed literature varies model capacity by attack phase. Second, the routing and MoE literature has solved capacity allocation but only on query-intrinsic signals, not on adversarial session dynamics. Third, the embedding and ATT&CK-labeling literature shows the phase signal is observable [16, 21, 49, 50], and the production deployment of Magazov *et al.* [1] shows it is reliable at scale.

The strands are individually mature and mutually disconnected—a disconnect the co-occurrence structure illustrates, with honeypot and model-routing terms not co-occurring across the fifty studies, and which the content-level sensitivity analysis of Section III-E confirms is not merely an artifact of the term-group search design. This is the niche the present work occupies: porting the cost–fidelity machinery of Section IV-C into the temporal frame of Section IV-B, instrumented by the analytic substrate of Section IV-D.

VI. Proposed Conceptual Framework

Relation to prior work and author disclosure: this section reports no new empirical results. It presents a conceptual framework whose components the preceding mapping synthesis identifies as jointly absent from the literature. To keep the review methodologically independent, we separate the framework's conceptual motivation, which is the contribution of this paper, from

its implemented and empirically evaluated realization, which the present authors report in separate primary studies. We disclose that those companion works share authorship with this review. One of them, the phase-adaptive model routing framework, has since been published [9] and is now cited formally; the quantitative figures in Section VI-C are its reported benchmark results, which the reader can verify independently at that reference, and not measurements produced by this review. The second, an unprivileged honeypot substrate for observing autonomous cyber agents, remains under peer review and is therefore described here by its operating principles only, with no quantitative claim attached to it.

This review reproduces neither study's results and derives its gap analysis solely from the fifty mapped works.

The framework turns the uniform LLM honeypot into a phase-adaptive, agent-oriented one by composing two layers, shown in Fig. 4. The first layer estimates the attacker's phase from the command stream and routes inference accordingly; the second observes the adversary at campaign granularity under a strict privilege boundary. Together they operationalize the three measurement requirements derived in Section V-A—phase-resolved fidelity, latency coupled to phase, and campaign-level observation.

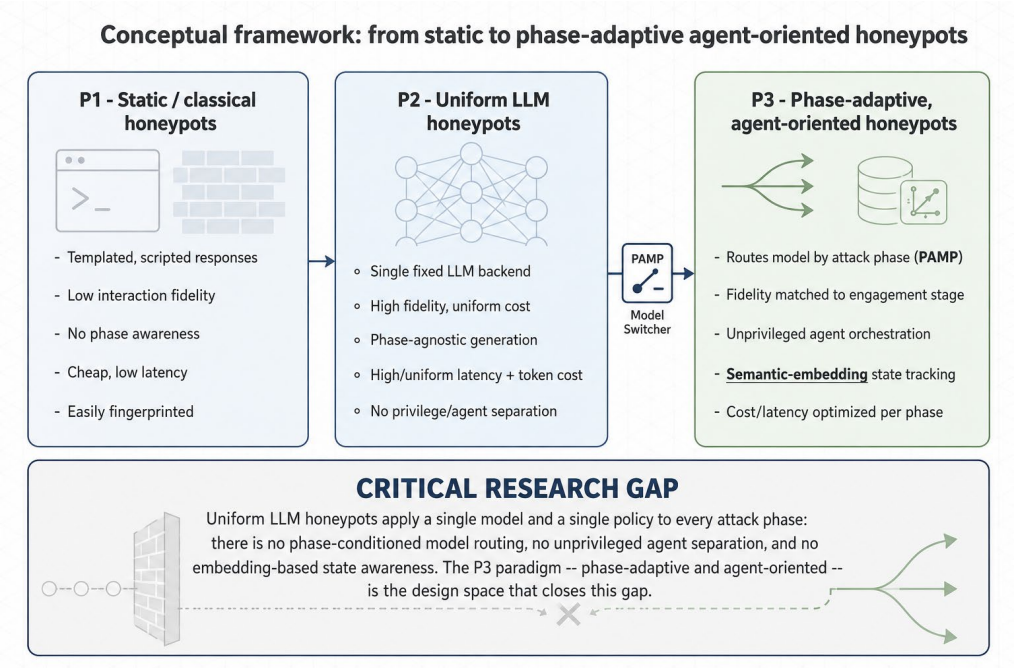


Fig. 4. The proposed phase-adaptive, agent-oriented honeypot. An embedding-based phase estimator drives an O(1) phase-to-model-tier dispatch layer (the routing foundation), composed with an unprivileged, campaign-level observation substrate (the agent-observation foundation).

A. Phase-Adaptive Model-Routing Layer

1) Phase estimation

Let $c(t)$ denote the t-th command issued by the attacker in a campaign, and let $E(\cdot)$ be a fixed sentence encoder that maps a command to a dense, d-dimensional embedding $E\big(c(t)\big)$. To suppress the noise of any single command, the estimator operates over a sliding window of up to the W most recent commands, summarized by their mean embedding

$$\overline{E}(t) \;=\; \frac{1}{\min(t,W)} \cdot \sum_i E\big(c(i)\big) \qquad (1)$$

The window length contracting to t until W commands have accrued. The attacker phase is treated as a latent categorical variable $\varphi \in \Phi = \{\text{Recon}, \text{Exploit}, \text{Persist}\}$, with each phase represented by a prototype vector $P(\varphi)$ of the same dimension d, estimated offline from labeled command exemplars. Affinity between two vectors is measured by cosine similarity,

$$\text{Sim}(u, v) \;=\; \frac{u \cdot v}{\|u\| \cdot \|v\|} \qquad (2)$$

The estimated phase at step t is the prototype most similar to the window embedding $\overline{E}(t)$,

$$\hat{\varphi}(t) \;=\; \underset{\varphi \,\in\, \Phi}{\text{argmax}}\ \text{Sim}\left(\overline{E}(t), P(\varphi)\right) \qquad (3)$$

The recoverability of $\hat{\varphi}(t)$ is not assumed but inherited from the embedding evidence of Section IV-D [16, 21, 49] and from the production-scale stage-recognition result of Magazov *et al.* [1].

We do not claim that phase boundaries are crisp. Real intrusions interleave phases: reconnaissance commands recur during exploitation, credential access and discovery straddle the boundary, and a sophisticated adversary may blur them deliberately. Accordingly $\hat{\varphi}(t)$ is best understood as a probabilistic estimate over a window rather than a hard segmentation. The design tolerates this in two ways. The window mean of Eq. (1) is explicitly a

smoothing device against single-command ambiguity, and because the router needs only the coarse low- against high-stakes distinction of the dispatch rule below, rather than an exact tactic, several ATT&CK tactics may co-occur within one routing phase without changing the dispatched tier.

Deliberate phase blurring is therefore an adversarial concern rather than a modeling error, and is treated by the hysteresis of Eq. (5) and by the defenses of Section VII-B. The empirical phase-estimation accuracy reported by the companion study [9] (macro-F1 of 0.87) is a laboratory benchmark, not evidence from live adversarial traffic, and we treat it as such.

2) O(1) dispatch under a fidelity constraint

Let $M = \{m_1, m_2, m_3\}$ be a pool of model tiers ordered by capability and cost, with capability rank $\rho\big(m(k)\big) = k$. For each phase the router selects the cheapest tier whose expected fidelity meets a phase-specific floor $\tau(\varphi)$:

$$D(\varphi) = \underset{m \in M}{\operatorname{argmin}} \ \mathrm{Cost}(m) \quad \text{s.t.} \quad \mathrm{Fid}(m, \varphi) \geq \tau(\varphi) \tag{4}$$

Because evaluating Eq. (4) requires scoring the 3×3 cost–fidelity matrix only once, and offline, the result is a static 3×3 dispatch table indexed by phase. Online inference is therefore a constant-time lookup $D\big(\hat{\varphi}(t)\big)$ with no per-turn optimization, which keeps the routing overhead negligible relative to generation latency. A representative solution sets a low fidelity floor for reconnaissance and high floors for exploitation and persistence, so Eq. (4) allocates the cheap tier m_1 to reconnaissance and reserves the capable tier m_3 for exploitation and persistence. The floors are calibrated so that the most capable tier m_3 satisfies every $\tau(\varphi)$, ensuring the feasible set in Eq. (4) is never empty and $D(\varphi)$ is always well defined.

TABLE IV. MAPPING OF THE ROUTING TAXONOMY TO MITRE ATT&CK TACTICS

Routing phase	MITRE ATT&CK tactics collapsed into it	Fidelity floor
Recon	Reconnaissance; Resource Development; Discovery	Low
Exploit	Initial Access; Execution; Privilege Escalation; Defense Evasion; Credential Access; Lateral Movement	High
Persist	Persistence; Command and Control; Collection; Exfiltration; Impact	High

Note: Cost-equivalence mapping from the three-phase routing taxonomy of Section VI-A to the fourteen Enterprise tactics of MITRE ATT&CK; tactic names follow the ATT&CK Enterprise matrix. The mapping is conceptual, not an empirical frequency distribution; the fidelity floor is the routing parameter $\tau(\varphi)$ of Eq. (4), stated qualitatively because its numeric value is a deployment-specific calibration.

The three-phase taxonomy $\Phi = \{\mathrm{Recon}, \mathrm{Exploit}, \mathrm{Persist}\}$ is deliberately coarser than the fourteen tactics of MITRE ATT&CK, and the reduction is driven by the routing decision rather than by forensic ambition. The router needs only enough resolution to separate low-stakes turns, where a cheap tier suffices, from high-stakes turns, where fidelity must be protected; a finer taxonomy would multiply the dispatch table without changing that cost logic, because most ATT&CK tactics collapse onto the same fidelity floor. The three phases are therefore cost-equivalence classes over ATT&CK tactics, as Table IV summarizes. The coarsening applies only to the routing key: the observation substrate of Section VI-B still labels every command with its full per-command ATT&CK technique, so credential access, discovery, collection, lateral movement and exfiltration remain available for analysis, and further phases can be added with no change to Eqs. (1)–(5).

3) Monotonically non-decreasing hysteresis

A naive application of Eq. (4) is vulnerable to a mimicry attack: an adversary aware of phase-conditioned routing could escalate to trigger a capable tier, then interleave stealthy or trivially simple commands to drive $\hat{\varphi}(t)$ back to Recon and downgrade the honeypot to a cheaper, less convincing model precisely when high fidelity matters most. To close this vector, the router does not serve $D\big(\hat{\varphi}(t)\big)$ directly. Instead it maintains a campaign-scoped escalation state $s(t)$ governed by a monotonically non-decreasing hysteresis mechanism,

$$s(t) = \max\Big(s(t-1), \rho\Big(D\big(\hat{\varphi}(t)\big)\Big)\Big), \quad s_0 = \rho(m_1) \tag{5}$$

and serves the tier $m(s)$. Since $s(t) \geq s(t-1)$ for all t, the served capability can only rise within a campaign: once exploitation or persistence has escalated the honeypot, no subsequent low-signal command can lower the tier. The state $s(t)$ is reset only at a campaign boundary, when a genuinely new adversary is observed. The mechanism trades a bounded amount of latency saving—capability is held at its high-water mark rather than tracking $\hat{\varphi}(t)$ downward—for robustness against deliberate phase spoofing, a trade we argue is correct in an adversarial setting where the cost of under-serving a real exploitation turn dominates the cost of over-serving a few reconnaissance turns.

To make Eqs. (1)–(5) concrete, consider a window of three commands: uname -a, wget http://x/p, chmod +x p. At $t = 1$ the window mean lies nearest the Recon prototype, so Eq. (3) gives $\hat{\varphi}$ = Recon, Eq. (4) dispatches the cheap tier m_1 and Eq. (5) sets $s = \rho(m_1)$. By $t = 3$ the window mean has shifted toward the Exploit prototype, since payload retrieval and execution now dominate the window, so $\hat{\varphi}$ = Exploit and Eq. (4) yields m_3; the hysteresis update Eq. (5) raises s from $\rho(m_1)$ to $\rho(m_3)$ and latches there. A subsequent innocuous ls would drive $\hat{\varphi}$ back to Recon, but because s is monotonic the served tier remains m_3, which is exactly the downgrade resistance Eq. (5) is designed to provide.

B. Unprivileged Agent-Observation Substrate

The second layer never grants root. Security-relevant commands are resolved through deterministic logic while the LLM fills only the benign long tail, so a captured session cannot be escalated into a usable foothold. Capability-gap canaries—hex decoding, inline arithmetic,

raw-byte recovery—fingerprint automated agents by exposing the narrow competencies that distinguish scripted tooling from interactive operators. Telemetry is recorded at the campaign level: inter-command timing, kill-chain transitions, and per-command MITRE ATT&CK labels, taking the campaign rather than the session as the unit of analysis. This substrate directly answers the three absences catalogued in Section V-A—it makes fidelity phase-attributable, couples latency accounting to the dispatched tier, and lifts observation from the session to the campaign.

C. Integration and Expected Properties

Composed, the two layers let a honeypot allocate inference by phase while observing agentic adversaries at campaign granularity. The routing layer is evaluated not with phase-blind averages but with two composite metrics—Phase-Weighted Fidelity (PWF) and the Latency-Savings Ratio (LSR)—so that efficiency gains are credited only when they preserve fidelity on the high-value turns. Both are defined operationally below; the definitions and the weighting scheme are those of the companion study [9], stated here in explicit algebraic form.

Phase-weighted fidelity is a value-weighted average of phase-wise fidelity. Let $f\big(c(t)\big) \in [0,1]$ be the per-command token-level fidelity of the honeypot response to command $c(t)$—the scale on which [9] reports PWF, although Eq. (6) is agnostic to the underlying per-command fidelity measure—and let $\varphi(t)$ be the ground-truth phase in which that command was issued. Writing $n(\varphi)$ for the number of benchmark commands issued in phase φ, $\overline{F}(\varphi)$ for their mean fidelity, and $w(\varphi) > 0$ for the value weight assigned to that phase, PWF is

$$\mathrm{PWF} = \frac{\sum_{\varphi \in \Phi} w(\varphi) \cdot n(\varphi) \cdot \overline{F}(\varphi)}{\sum_{\varphi \in \Phi} w(\varphi) \cdot n(\varphi)} \tag{6}$$

Eq. (6) is a weighted mean, so PWF is bounded on the same [0,1] scale as the fidelity measure underlying it and is comparable across systems only when the weight vector is reported alongside the value. The weights encode operational value—the cost of an unconvincing response in each phase—and deliberately not occurrence frequency. Frequency is already carried by $n(\varphi)$, so the uniform choice $w(\varphi) = 1$ for every φ reduces Eq. (6) exactly to the conventional corpus-level fidelity average, which a honeypot workload dominated by reconnaissance renders insensitive to precisely the turns that matter; weighting by frequency a second time would compound that insensitivity rather than correct it. The companion study [9] accordingly sets $w(\mathrm{Recon}) = 1$ and $w(\mathrm{Exploit}) = w(\mathrm{Persist}) = 2$, on the ground that an inconsistent response after a foothold has been established is substantially more likely to expose the decoy than an inconsistent response during initial enumeration. The 2:1 ratio is a declared parameter of the metric rather than a fitted quantity, and any study reporting PWF should state the weight vector with it, since a PWF value is uninterpretable without one.

The latency-savings ratio expresses the relative reduction in latency achieved with respect to a uniform-model baseline. Writing $\overline{L}(\mathrm{PA})$ for the mean response latency of the phase-adaptive configuration and $\overline{L}(b)$ for that of a reference uniform baseline b, both measured over the same benchmark,

$$\mathrm{LSR}(b) = \frac{\overline{L}(b) - \overline{L}(\mathrm{PA})}{\overline{L}(b)} \tag{7}$$

LSR is reported as a proportion or, equivalently, as a percentage, with positive values denoting a saving. Because it is defined relative to a baseline it is meaningful only when that baseline is named, and we name the baseline with every value we quote. To keep the measurement basis of Eqs. (6) and (7) uniform, all quantities entering them are computed over the same 412-session benchmark under identical session ordering and identical random seeds; latency is wall-clock response latency measured per command; and the means $\overline{L}(\cdot)$ are taken over five independent runs with backend warm-up, following the protocol of Magazov *et al.* [9]. On that protocol the companion study reports LSR = 0.38 against the Application-Programming-Interface (API)-hosted high-capacity uniform baseline (3,168 ms) and LSR = 0.45 against the locally hosted mid-capacity baseline (3,584 ms); against its small-model uniform baseline (492 ms) LSR is negative, which is exactly why an LSR figure is reported only in conjunction with the fidelity that Eq. (6) measures and never on its own.

The companion study of Magazov *et al.* [9] provides an empirical picture of both quantities. On a controlled, reproducible benchmark of 412 stage-annotated command sessions it reports that the design lowers mean response latency from 3,168 ms for an API-hosted high-capacity uniform baseline to 1,974 ms, a reduction of about 38%, and about 45% against a locally hosted mid-capacity baseline at 3,584 ms. It holds token-level fidelity close to the high-capacity baseline (cosine similarity 0.39 against 0.40) and phase-weighted fidelity likewise close to it (PWF 0.49 against 0.51, on the weights given above), while maintaining an online stage-classification macro-F1 of 0.87. Read together, the two metrics state the property the framework claims: most of the latency of a uniformly high-capacity honeypot is removed, and the fidelity that survives is concentrated on the phases weighted highest.

The same study reports that the hysteresis of Section VI-A neutralizes the downgrade-by-mimicry attack at a small, bounded cost in LSR, with results stable under five-fold stratified group cross-validation. These are laboratory-benchmark results for the routing layer alone; validating the integrated two-layer system against live adversarial traffic remains the first item of the agenda below.

One evaluation axis that phase-weighted fidelity does not by itself capture is persona consistency across tier switches: whether an attacker perceives a single coherent operator even as the honeypot changes model tier mid-session. The honeypot-evaluation metrics and persona-shaping techniques of Junnuthula and Singh [52]

begin to formalize exactly this property, and we flag it as a first-class evaluation requirement specific to routed honeypots, alongside PWF and LSR.

VII. Open Challenges, Limitations, and Future Directions

A. Evaluation under Phase-Adaptive Inference

The corpus shows fidelity is rarely reported per phase. Phase-resolved metrics—the phase-weighted fidelity and latency-savings ratio defined in Eqs. (6) and (7)—are needed so phase-adaptive systems are not penalized or flattered by phase-blind averaging. Establishing such metrics requires phase-labeled interaction traces, which the field currently lacks: only the embedding and CTI strand labels commands with MITRE ATT&CK techniques [21], and no public honeypot dataset carries per-turn phase annotations. A shared, phase-labeled benchmark would let phase-adaptive and uniform systems be compared on the axis that distinguishes them, and would let latency savings be reported against a fidelity floor rather than in isolation.

B. Adversarial Robustness

Phase estimation and any retrieval component widen the attack surface, and a phase-adaptive honeypot must treat its own phase classifier and retrieval store as adversarially reachable [43, 46]. The hysteresis mechanism of Section VI-A neutralizes the specific downgrade-by-mimicry attack, but it does not address the symmetric risk of forced over-escalation, in which an attacker cheaply induces the capable tier to exhaust inference budget, nor does it harden the prototype vectors $P(\varphi)$ themselves against poisoning. Forced over-escalation deserves separate treatment, because it is the direct price of the monotonicity that defeats the downgrade attack: an adversary who front-loads a single complex command can pin the session to the most expensive tier for its entire remaining length. We therefore treat over-escalation as a first-class design constraint and set out four complementary mitigations, each of which reintroduces a controlled, non-monotone relaxation without reopening the downgrade vector.

First, a per-campaign inference-budget cap bounds the total number of high-tier invocations, after which the router falls back to a mid tier and flags the session, converting an unbounded cost into a bounded and observable one. Second, a cooldown or decay term permits de-escalation after a sustained run of low-signal, low-complexity turns—that is, only once the evidence for a genuine high-stakes phase has decayed—so a single early escalation cannot dominate an arbitrarily long session while a short burst of real exploitation still latches capability. Third, rate-limiting on escalation triggers caps how quickly and how often a session may climb tiers, blunting a preloaded escalation. Fourth, anomaly monitoring on escalation dynamics—escalation frequency, cost per session, and the ratio of high-tier turns to observed exploitation evidence—surfaces sessions whose escalation profile is inconsistent with a real intrusion.

Each trades a measurable increase in downgrade exposure for reduced over-escalation cost; quantifying that trade-off requires the benchmark this review calls for and is left as future work. Robust prototype estimation against poisoning of $P(\varphi)$ remains an open problem in its own right.

C. Agentic Adversaries, Calibration, and Ethics

As autonomous attackers proliferate [8, 53], honeypots double as measurement instruments for agent behavior, and incident-response calibration under adversarial evidence [45] becomes a deployment concern. Engagement-prolonging deception also carries ethical and legal constraints that production deployments must document: entrapment boundaries, data-handling obligations for captured attacker artifacts, and containment guarantees so a high-interaction decoy cannot be turned into a launch point. The unprivileged substrate of Section VI-B is partly a response to this last concern, but the broader governance of LLM-driven deception—who may run it, against whom, and with what disclosure—remains underdeveloped relative to the speed of deployment.

D. Limitations of This Review

The purposive top-50 cap privileges influence over completeness; lower-velocity but relevant work is necessarily excluded, and citation velocity favors older works within each cohort despite the dual-cohort correction. Index-resolved citation counts inherit index coverage gaps, and three included works carry no DOI because their venues had not yet registered one at the search date. The corpus also leans on preprints: 23 of the 50 studies are not yet archival by venue of record, several of them from 2026 and not yet widely vetted, so part of the evidence base may shift as those works complete peer review. We mitigate this by recording the version of record wherever one had appeared and by reporting the split explicitly (Sections III-F and IV-A).

Finally, the proposed framework is presented here as a conceptual architecture. Its routing layer has since received a benchmark evaluation in the companion study [9], but that evaluation covers the routing layer alone under laboratory conditions; the integrated two-layer system and its validation against live adversarial traffic remain the primary evaluation called for above.

VIII. Conclusion

Across fifty influential studies, honeypot adaptivity has matured at the policy and generation layers, and cost-aware routing has matured the allocation of model capacity—but in the literature we survey the two have not met, and no LLM honeypot in that literature—including the broader pool screened in the sensitivity analysis of Section III-E—varies its inference by attack phase. The phase signal is observable from command semantics and reliable in production. We formalized this gap and proposed a phase-adaptive, agent-oriented framework that couples an embedding-based phase estimator and an O(1) phase-to-model dispatch layer, hardened by a

monotonically non-decreasing hysteresis mechanism, with an unprivileged, campaign-level observation substrate. The trajectory the corpus traces—from static, to uniform LLM, to phase-adaptive honeypots—is, on this evidence, the field's most consequential near-term direction.

The argument the review assembles is ultimately one of convergence. The deception community has the workload and the phase signal but not the allocation machinery; the routing community has the allocation machinery but not the adversarial signal; and the embedding community has shown the signal is recoverable and labelable. None of the three needs a fundamental advance for the synthesis to proceed—what is missing is the integration, and the framework of Section VI is a concrete proposal for it. The immediate work is empirical: build the phase-labeled benchmark Section VII calls for, instrument a deployment with the proposed dispatch layer, and measure phase-weighted fidelity and latency savings against a uniform baseline. Until those measurements exist the framework remains a hypothesis, but it is a hypothesis the reviewed evidence makes both testable and, we argue, overdue.

APPENDIX: PER-STUDY DATA EXTRACTION

TABLE AI. FULL DATA EXTRACTION FOR THE 50 INCLUDED STUDIES

Ref. No.	Study	Honeypot Type	LLM Backend	Core Contribution
[1]	Magazov *et al.* (2026)	SSH (LLM, high-interaction)	fine-tuned LLM + embeddings	Published anchor: LLM SSH honeypot with embedding-based command-stage analytics over 6-month/4M-event deployment
[2]	Bridges *et al.* (2025)	survey (honeypots+LLMs)	n/a (SoK)	SoK synthesizing LLM-powered honeypots; confirms adaptive inference absent
[3]	Javadpour *et al.* (2024)	survey (deception)	n/a	Comprehensive survey of cyber-deception techniques to improve honeypot performance
[4]	Purnama *et al.* (2025)	SLR (AI honeypots)	n/a (PRISMA)	PRISMA systematic review of AI-enhanced honeypots for threat detection
[5]	Azzahri *et al.* (2024)	SSH/Telnet (Cowrie)	none (static)	Cowrie deployment analysis of SSH/Telnet attacks
[6]	Chen *et al.* (2023)	n/a (LLM cascade)	LLM cascade	FrugalGPT: cost-efficient LLM cascade (prompt adaptation, approximation, cascade)
[7]	Ong *et al.* (2024)	n/a (LLM routing)	router (strong/weak LLM)	RouteLLM: learning to route between strong/weak LLMs from preference data
[8]	Xu *et al.* (2025)	survey (LLM agents/attacks)	n/a (survey)	Survey of LLM-based agents in autonomous cyberattacks
[11]	Ding *et al.* (2024)	n/a (LLM routing)	router (small+large)	Hybrid LLM: cost-efficient, quality-aware query routing (ICLR)
[12]	Muennighoff *et al.* (2024)	n/a (MoE LLM)	OLMoE (1B active/7B)	OLMoE: open mixture-of-experts LLM with sparse routing
[13]	Jitkrittum *et al.* (2025)	n/a (LLM routing)	universal router	Universal model routing for efficient LLM inference
[14]	Zhang *et al.* (2025)	n/a (LLM routing)	RL-trained router	Router-R1: RL for multi-round LLM routing and aggregation
[15]	Wang *et al.* (2025)	n/a (LLM routing)	dynamic mixed-LLM router	MixLLM: dynamic routing across mixed LLMs
[16]	Koukoulis *et al.* (2025)	n/a (IDS embeddings)	transformer embeddings	Self-supervised transformer contrastive learning for IDS
[17]	Sladic *et al.* (2023)	SSH (LLM, medium-interaction)	GPT-3.5/4 (shelLM)	shelLM: session-stateful generative SSH/Linux shell honeypot
[18]	Mohammadshahi *et al.* (2024)	n/a (LLM routing)	router (Routoo)	Routoo: learning to route among diverse LLMs effectively under cost
[19]	Otal *et al.* (2024)	SSH/Telnet (LLM, interactive)	fine-tuned Llama-3	LLMs as advanced interactive honeypot systems
[20]	Shirkavand *et al.* (2025)	n/a (LLM routing)	contrastive router	Cost-aware contrastive routing for LLMs
[21]	Lekssays *et al.* (2025)	n/a (CTI + RAG)	RAG + LLM re-ranking	TechniqueRAG: RAG for ATT&CK technique annotation in CTI text
[22]	Guan *et al.* (2023)	IoT (high-interaction, RL)	none (RL policy)	HoneyIoT: RL-driven adaptive high-interaction IoT honeypot
[23]	Naseri *et al.* (2026)	SSH (adaptive, Cowrie+RL)	none (Q-learning/MDP)	Q-Cowrie: self-learning adaptive SSH honeypot (Q-learning + MDP)
[24]	Veluchamy *et al.* (2021)	DoS-resistant (DRL)	none (deep RL)	Deep RL for building honeypots against runtime DoS
[25]	Hammar *et al.* (2021)	defensive (optimal stopping)	none (RL/game theory)	Intrusion prevention via optimal stopping / RL
[26]	Malhotra (2025)	SSH (LLM, real-time)	13 backends (Gemma/Phi/etc.)	LLMHoney: real-time SSH honeypot, latency-fidelity curve across 13 models
[27]	Vasilatos *et al.* (2024)	ICS/PLC (LLM)	fine-tuned LLM	LLMPot: LLM-based ICS protocol & physical-process emulation
[28]	Wang *et al.* (2024)	terminal (LLM)	ChatGPT (structured prompts)	HoneyGPT: structured-prompt LLM terminal honeypot
[29]	Sladic *et al.* (2025)	multi-protocol (LLM)	multiple LLMs	VelLMes: high-interaction AI-based deception framework (SSH/HTTP/etc.)
[30]	Jiménez-Román *et al.* (2025)	LDAP (LLM)	LLM-based	Intelligent LLM-based LDAP honeypot

[31]	Guan *et al.* (2024)	shell (LLM)	LLM-based	HoneyLLM: enabling shell honeypots with large language models
[32]	Kahlhofer *et al.* (2025)	Kubernetes (deception orchestration)	n/a (orchestration)	Koney: cyber-deception orchestration framework for Kubernetes
[33]	Reworr *et al.* (2024)	SSH (LLM, agent-monitoring)	prompt-injection instrumented	LLM Agent Honeypot: monitoring autonomous AI hacking agents in the wild
[34]	Qiu *et al.* (2024)	n/a (MoE routing)	recurrent router	Layerwise recurrent router for mixture-of-experts (ICLR)
[35]	Sun *et al.* (2026)	n/a (MoE routing)	threshold router	Expert threshold routing with dynamic computation allocation
[36]	Li *et al.* (2025)	n/a (expert routing)	low-rank expert adapters	Ensembles of low-rank expert adapters with routing
[37]	Liu *et al.* (2025)	n/a (MoE routing)	LLM-as-router	LLMoE: LLM-based gating/routing in mixture-of-experts
[38]	Moslem *et al.* (2026)	survey (LLM routing)	n/a (survey)	Survey of dynamic model routing & cascading for efficient LLM inference
[39]	Garg *et al.* (2026)	n/a (LLM routing)	hybrid dynamic router	HyDRA: hybrid dynamic routing for heterogeneous LLM pools
[40]	Mitra *et al.* (2025)	n/a (LLM CTI IDS rules)	LLM CTI miner	FALCON: autonomous CTI mining with LLMs for IDS rule generation
[41]	Bertiger *et al.* (2025)	n/a (LLM detection rules)	LLM rule generator	Evaluating LLM-generated detection rules in cybersecurity
[42]	Alhuzali (2025)	n/a (CTI + RAG)	RAG LLM	LLM-powered threat intelligence via retrieval-augmented generation
[43]	Meng *et al.* (2025)	n/a (LLM CTI)	LLM-assisted CTI	Uncovering vulnerabilities of LLM-assisted cyber threat intelligence
[44]	Liu *et al.* (2025)	n/a (multi-agent IR + RAG)	RAG multi-agent	AutoBnB-RAG: RAG-enhanced multi-agent incident response
[45]	Barnes (2026)	n/a (IR agent eval)	LLM IR agent	OpenSec: measuring incident-response agent calibration
[46]	Xu *et al.* (2026)	survey (RAG security)	n/a (taxonomy)	Securing RAG: taxonomy of attacks, defenses, directions
[47]	Chhetri *et al.* (2025)	n/a (attack prediction)	transformer	Transformer models for predicting consequences of cyber attacks
[48]	Mahdaouy *et al.* (2026)	n/a (NetFlow IDS)	deep learning embeddings	Deep learning for contextualized NetFlow-based intrusion detection
[49]	Othman *et al.* (2025)	n/a (vuln mapping)	sentence transformer	Sentence-transformer mapping of attack descriptions to vulnerabilities
[50]	Liu *et al.* (2025)	n/a (IoT IDS)	self-supervised embeddings	IoT/sensor IDS via semantic analysis and self-supervised embeddings
[51]	Biswas *et al.* (2026)	n/a (explainable IDS)	encoder LLM	Attribution-driven explainable IDS with encoder-based LLMs
[53]	Heckel *et al.* (2024)	n/a (LLM defense)	defensive LLM agents	Countering autonomous cyber threats with defensive LLM agents

Note: Complete per-study extraction for all 50 included studies, in the schema of Table I, taken by the authors from the full texts with bibliographic fields index-resolved as described in Section III-D. "n/a" marks a field that does not apply to the study; any field the index could not return was left blank and flagged rather than estimated. The aggregate counts derived from this table appear in Table III. Abbreviations: DoS, denial of service; DRL, deep reinforcement learning; IDS, intrusion-detection system; IR, incident response; RL, reinforcement learning; SLR, systematic literature review.

Conflict of Interest

The authors declare no conflict of interest.

Author Contributions

R.M. served as the principal investigator, conceived the overarching study, led the systematic mapping review, designed the phase-adaptive model routing framework, and wrote the primary manuscript; F.U. coordinated the research methodology, validated the mapping review results, and refined the formal gap analysis; K.A. conducted the title and abstract screening, managed duplicate removal, and executed the age-normalized citation velocity ranking; B.Y. performed data extraction according to the study schema and generated the bibliometric and keyword co-occurrence trend analyses; Zh.A. audited the PRISMA flow diagram funnel, evaluated full-text eligibility against the selection criteria, and contributed to the discussion on embedding-based attack analysis; Y.Sh. contributed to the database search execution, assisted in the architectural formulation of the phase-to-model dispatch layer, and supported the data-extraction integrity protocol; all authors participated in the critical revision of the manuscript and approved the final version.

Acknowledgment

The authors wish to thank the anonymous reviewers for their valuable comments and constructive suggestions that helped improve the quality of this manuscript. The authors also express their gratitude to Al-Farabi Kazakh National University, Satbayev University, and Almaty Management University for providing the institutional support and research infrastructure necessary to conduct this study.

References

[1] R. Magazov, K. Abeshev, F. Uralova, and G. Aksholak, "Command Pattern Recognition in Linux-Based Cyber Incidents Using Large Language Models for Intelligent Honeypot Systems," in *Proc. 2026 18th International Conference on Electronics, Computer, and Computation (ICECCO)*, 2026.

[2] R. A. Bridges, T. Mitchell, M. Muñoz, and T. Henriksson, "SoK: Honeypots & LLMs, more than the sum of their parts?" arXiv preprint, arXiv:2510.25939, 2025.

[3] 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*, 2024. doi: 10.1016/j.cose.2024.103792

[4] L. H. Purnama and D. H. Prasetyo, "Effectiveness of artificial intelligence-based adaptive honeypots in cyber threat detection: A systematic literature review and meta-analysis," *Jurnal Sisfokom (Sistem Informasi dan Komputer)*, 2025. doi: 10.32736/sisfokom.v14i4.2403

[5] M. N. Azzahri *et al.*, "The Application of Cowrie Honeypot to Analyze Attacks on SSH and Telnet Protocols," in *Proc. 2024 IEEE 2nd International Conference on Electrical Engineering, Computer and Information Technology (ICEECIT),* 2024. doi: 10.1109/ICEECIT63698.2024.10859786

[6] L. Chen, M. Zaharia, and J. Y. Zou, "FrugalGPT: How to use large language models while reducing cost and improving performance," *Trans. Mach. Learn. Res*., 2023.

[7] I. Ong *et al*., "RouteLLM: Learning to Route LLMs with Preference Data," in *Proc. International Conference on Learning Representations*, 2025.

[8] M. Xu *et al.*, "Forewarned is Forearmed: A Survey on Large Language Model-based Agents in Autonomous Cyberattacks," arXiv preprint, arXiv:2505.12786, 2025.

[9] R. Magazov, F. Uralova, K. Abeshev, G. Akhmedi, and G. Aksholak, "Phase-Adaptive Model Routing in LLM-Driven SSH Honeypots: Balancing Response Fidelity and Latency Across the Attack Lifecycle," *Future Internet*, vol. 18, no. 7, 359, 2026. doi: 10.3390/fi18070359

[10] U. Salviati, F. De Gaspari, M. Conti, and L. V. Mancini, "ShellGames: Speculative LLM-Driven SSH Deception," arXiv preprint, arXiv:2606.17986, 2026.

[11] D. Ding *et al.*, "Hybrid LLM: Cost-Efficient and Quality-Aware Query Routing," in *Proc. International Conference on Learning Representations*, 2024.

[12] N. Muennighoff *et al*., "OLMoE: Open Mixture-of-Experts Language Models," arXiv preprint, arXiv:2409.02060, 2024.

[13] W. Jitkrittum *et al*., "Universal Model Routing for Efficient LLM Inference," arXiv preprint, arXiv:2502.08773, 2025.

[14] H. Zhang, T. Feng, and J. You, "Router-R1: Teaching LLMs Multi-Round Routing and Aggregation via Reinforcement Learning," arXiv preprint, arXiv:2506.09033, 2025.

[15] X. Wang *et al.,* "MixLLM: Dynamic Routing in Mixed Large Language Models," in *Proc. North American Chapter of the Association for Computational Linguistics*, 2025. doi: 10.18653/v1/2025.naacl-long.545

[16] I. Koukoulis, I. Syrigos, and T. Korakis, "Self-Supervised Transformer-based Contrastive Learning for Intrusion Detection Systems," arXiv preprint, arXiv:2505.08816, 2025.

[17] M. Sladic, V. Valeros, C. Catania, and S. García, "LLM in the Shell: Generative Honeypots," in *Proc. 2024 IEEE European Symposium on Security and Privacy Workshops (EuroS&PW)*, 2023. doi: 10.1109/EuroSPW61312.2024.00054

[18] A. Mohammadshahi, A. Shaikh, and M. Yazdani, "Routoo: Learning to Route to Large Language Models Effectively," arXiv preprint, arXiv:2401.13979, 2024.

[19] H. T. Otal and M. A. Canbaz, "LLM Honeypot: Leveraging Large Language Models as Advanced Interactive Honeypot Systems," in *Proc. IEEE Conference on Communications and Network Security*, 2024. doi: 10.1109/CNS62487.2024.10735607

[20] R. Shirkavand, S. Gao, P. Yu, and H. Huang, "Cost-Aware Contrastive Routing for LLMs," arXiv preprint, arXiv:2508.12491, 2025.

[21] A. Lekssays, U. Shukla, H. Sencar, and Md. R. Parvez, "TechniqueRAG: Retrieval Augmented Generation for Adversarial Technique Annotation in Cyber Threat Intelligence Text," in *Proc. Annual Meeting of the Association for Computational Linguistics*, 2025. doi: 10.18653/v1/2025.findings-acl.1076

[22] C. Guan, H. Liu, G. Cao, S. Zhu, and T. L. L. Porta, "HoneyIoT: Adaptive High-Interaction Honeypot for IoT Devices Through Reinforcement Learning," *Wireless Network Security*, 2023. doi: 10.1145/3558482.3590195

[23] M. V. Naseri, I. Welch, J. Haseeb, and M. Mansoori, "Q-Cowrie: An adaptive honeypot to analyse attackers' behaviour," *International Journal of Information Security*, 2026. doi: 10.1007/s10207-026-012215

[24] S. Veluchamy and R. S. Kathavarayan, "Deep reinforcement learning for building honeypots against runtime DoS attack," *International Journal of Intelligent Systems*, 2021. doi: 10.1002/int.22708

[25] K. Hammar and R. Stadler, "Intrusion Prevention Through Optimal Stopping," *IEEE Transactions on Network and Service Management*, 2021. doi: 10.1109/TNSM.2022.3176781

[26] P. Malhotra, "LLMHoney: A Real-Time SSH Honeypot with Large Language Model-Driven Dynamic Response Generation," arXiv preprint, arXiv:2509.01463, 2025.

[27] C. Vasilatos, D. Mahboobeh, H. Lamri, M. Alam, and M. Maniatakos, "LLMPot: Dynamically Configured LLM-based Honeypot for Industrial Protocol and Physical Process Emulation," in *Proc. European Symposium on Security and Privacy*, 2024. doi: 10.1109/EuroSP63326.2025.00059

[28] Z. Wang *et al.*, "HoneyGPT: Breaking the Trilemma in Terminal Honeypots with Large Language Model," arXiv preprint, arXiv:2406.01882, 2024.

[29] M. Sladic, V. Valeros, C. Catania, and S. García, "VelLMes: A High-Interaction AI-Based Deception Framework," in *Proc. 2025 IEEE European Symposium on Security and Privacy Workshops (EuroS&PW)*, 2025. doi: 10.1109/EuroSPW67616.2025.00082

[30] J. Jiménez-Román, F. Almenares-Mendoza, and A. Sánchez-Macián, "Design and Development of an Intelligent LLM-based LDAP Honeypot," arXiv preprint, arXiv:2509.16682, 2025.

[31] C. Guan, G. Cao, and S. Zhu, "HoneyLLM: Enabling Shell Honeypots with Large Language Models," in *Proc. IEEE Conference on Communications and Network Security*, 2024. doi: 10.1109/CNS62487.2024.10735663

[32] M. Kahlhofer, M. Golinelli, and S. Rass, "Koney: A Cyber Deception Orchestration Framework for Kubernetes," in *Proc. 2025 IEEE European Symposium on Security and Privacy Workshops (EuroS&PW)*, 2025. doi: 10.1109/EuroSPW67616.2025.00084

[33] V. Reworr and D. Volkov, "LLM Agent Honeypot: Monitoring AI Hacking Agents in the Wild," arXiv preprint, arXiv:2410.13919, 2024.

[34] Z. Qiu *et al.*, "Layerwise Recurrent Router for Mixture-of-Experts," in *Proc. International Conference on Learning Representations*, 2025.

[35] H. Sun, Y. Liu, Y. Wu, and L. Sun, "Expert Threshold Routing for Autoregressive Language Modeling with Dynamic Computation Allocation and Load Balancing," arXiv preprint, arXiv:2603.11535, 2026.

[36] Y. Li, V. Gao, C. Zhang, and M. Torkamani, "Ensembles of Low-Rank Expert Adapters," in *Proc. International Conference on Learning Representations*, 2025.

[37] K. Liu and M. Lo, "LLM-Based Routing in Mixture of Experts: A Novel Framework for Trading," arXiv preprint, arXiv:2501.09636, 2025.

[38] Y. Moslem and J. D. Kelleher, "Dynamic Model Routing and Cascading for Efficient LLM Inference: A Survey," arXiv preprint, arXiv:2603.04445, 2026.

[39] A. Garg, S. Roy, J. Jang, F. Brancasi, and S. Fu, "HyDRA: Hybrid Dynamic Routing Architecture for Heterogeneous LLM Pools," arXiv preprint, arXiv:2605.17106, 2026.

[40] S. Mitra *et al*., "FALCON: Autonomous Cyber Threat Intelligence Mining with LLMs for IDS Rule Generation," arXiv preprint, arXiv:2508.18684, 2025.

[41] A. Bertiger *et al*., "Evaluating LLM Generated Detection Rules in Cybersecurity," arXiv preprint, arXiv:2509.16749, 2025.

[42] A. Alhuzali, "LLM-powered threat intelligence: A retrieval-augmented generation approach for cyber attack investigation," *PeerJ Computer Science*, 2025. doi: 10.7717/peerj-cs.3371

[43] Y. Meng *et al.*, "Uncovering Vulnerabilities of LLM-Assisted Cyber Threat Intelligence," arXiv preprint, arXiv:2509.23573, 2025.

[44] Z. Liu and A. Anwar, "AutoBnB-RAG: Enhancing Multi-Agent Incident Response with Retrieval-Augmented Generation," in *Proc. 2025 IEEE International Conference on Data Mining Workshops (ICDMW)*, 2025. doi: 10.1109/ICDMW69685.2025.00141

[45] J. Barnes, "OpenSec: Measuring Incident Response Agent Calibration Under Adversarial Evidence," arXiv preprint arXiv:2601.21083, 2026.

[46] Y. Xu *et al*., "Securing Retrieval-Augmented Generation: A Taxonomy of Attacks, Defenses, and Future Directions," arXiv preprint, arXiv:2604.08304, 2026.

[47] B. Chhetri and A. Namin, "The Application of Transformer-Based Models for Predicting Consequences of Cyber Attacks," in *Proc. Annual International Computer Software and Applications Conference*, 2025, doi: 10.1109/COMPSAC65507.2025.00074.

[48] A. E. Mahdaouy, I. A. Yahia, S. Oualil, and I. Berrada, "Deep Learning for Contextualized NetFlow-Based Network Intrusion Detection: Methods, Data, Evaluation and Deployment," arXiv preprint, arXiv:2602.05594, 2026.

[49] R. Othman, D. Rimawi, B. Rossi, and B. Russo, "From Attack Descriptions to Vulnerabilities: A Sentence Transformer-Based Approach," arXiv preprint, arXiv:2509.02077, 2025.
[50] Y. Liu and Y. Guo, "Enhancing Intrusion Detection for IoT and Sensor Networks Through Semantic Analysis and Self-Supervised Embeddings," in *Proc. Italian National Conference on Sensors*, 2025. doi: 10.3390/s25227074
[51] U. Biswas, S. Hasan, S. M. Farhan, N. Pillai, and C. Gudla, "Attribution-Driven Explainable Intrusion Detection with Encoder-Based Large Language Models," in *Proc. SoutheastCon*, 2026. doi: 10.1109/SoutheastCon63549.2026.11476386
[52] S. K. R. Junnuthula and V. K. Singh, "LLM-Based Cybersecurity Honeypots: Evaluation Metric Development and Persona Shaping via Structured Prompt Engineering," in *Proc. IEEE Int. Conf. Smart Technologies (SMART)*, 2025, doi: 10.1109/SMART66937.2025.11389636
[53] K. Heckel and A. Weller, "Countering Autonomous Cyber Threats," arXiv preprint, arXiv:2410.18312, 2024.

Copyright © 2026 by the authors. This is an open access article distributed under the Creative Commons Attribution License which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited (CC BY 4.0).