

Chameleon: An Adaptive AI-Driven Honey-pot Architecture Using Threat-Calibrated Particle Swarm Optimization and Semantic Deception Rapidly-Exploring Random Trees

Rohit Swami, Tushar Singh, Akash Warde, and Sri Muthu
Dept. of AI and Data Science, SIES Graduate School of Technology, Mumbai, India
rohitsaids124@gst.sies.edu.in, tusharnsaids124@gst.sies.edu.in,
akashkwads124@gst.sies.edu.in, srimumthulakshmananraids124@gst.sies.edu.in
Guide: Dr. Neethu Anna Sabu (neethus@sies.edu.in)

Abstract—An invariant behavioral profile is the defining vulnerability of traditional honeypot installations: a skilled adversary can confirm the presence of a deception environment within only a few diagnostic commands, limiting its intelligence value. High-cost commercial deception products (USD 100,000–150,000 per year) share a related weakness in that their response engines are not coupled to real-time model-driven feedback. Chameleon is an openly distributed adaptive honeypot platform introduced here to address both shortcomings. Three core components are integrated: a bidirectional long short-term memory (BiLSTM) classifier achieving 99.61% accuracy across seven threat categories at approximately two milliseconds CPU latency; a locally deployed Qwen3.5-0.8B language model (Qwen Team, 2026; Unsloth, 2026) delivering 90% contextual generation accuracy at 4.5 milliseconds average latency; and two domain-specific meta-heuristic engines. Threat-Calibrated Particle Swarm Optimization (TC-PSO) dynamically reshapes swarm inertia and objective amplification in proportion to the classifier’s anomaly output, enabling real-time adjustment of connection-holding delays. Semantic Deception Rapidly-Exploring Random Trees (S-RRT) drives deception schema evolution via exponentially scaled pheromone updates derived from a language-model severity assessment, while a depth-decay multiplier enforces a finite memory footprint. Across five benchmark runs (seeds 42–46), TC-PSO outperformed standard PSO by 48.1% in mean fitness (2.60 to 3.85) with a 32.7% convergence gain, and S-RRT exceeded standard RRT by 258.9% in best-run fitness (450.2 to 1,615.8), achieving a 329.2% gain at critical severity and a 24.9% memory reduction ($p < 0.01$). Operating costs are approximately USD 17 per month, a roughly 490-fold reduction versus commercial alternatives.

Index Terms—Adaptive honeypot; Threat-calibrated particle swarm optimization; Semantic deception RRT; BiLSTM classification; Meta-heuristic optimization; Deception technology

1. INTRODUCTION

Alert fatigue has become a structural liability within modern security operations centers. Industry reporting indicates that individual organizations process in excess of eleven thousand security events daily, yet triage capacity is so constrained that roughly seventy percent of those events receive no analyst attention (Ponemon Institute, 2023; Apruzzese et al., 2023). Perimeter-oriented defenses—next-generation firewalls,

signature-driven intrusion detection, and endpoint platforms—are architecturally ill-suited to this environment because each depends on previously catalogued threat patterns and can offer only reactive, post-event visibility against living-off-the-land movement or zero-day exploitation.

Deception infrastructure has long been proposed as a means of redistributing that asymmetry. Attacker engagement environments that appear operationally genuine allow defenders to observe adversarial tactics, techniques, and procedures in forensic detail while the adversary remains unaware that the session is controlled. The problem is that the dominant openly available sensors—Cowrie, a configurable SSH/Telnet emulation platform (Cowrie Project, 2023), and T-Pot, a consolidated multi-sensor aggregator (T-Pot Community, 2023)—share a common structural deficiency: their behavioral output is determined at deployment time and cannot be revised in response to the evolving session. As few as one or two probe commands, such as a `uname -a` query, may be sufficient for an informed attacker to match the response pattern against a known honeypot fingerprint and withdraw. Commercially available alternatives such as Illusive Networks and Thinkst Canary address this deficiency only partially, and none connects response generation to a live machine-learning pipeline. Market valuations underscore the scale of the unmet need: the deception technology sector was estimated at approximately 2.7 billion US dollars in 2024 and is projected to reach between 5.2 and 6.0 billion US dollars by 2029 (MarketsandMarkets, 2024), with open-source capability lagging well behind the commercial frontier.

Chameleon is introduced here as a production-complete, MIT-licensed adaptive honeypot designed to close this capability gap. Five primary contributions are advanced. First: TC-PSO, which, to the authors’ knowledge, represents the inaugural application of a BiLSTM-derived anomaly signal to govern particle inertia and objective amplification within a honeypot tarpit-delay optimization context. Second: S-RRT, the first variant of the rapidly-exploring random tree family in which pheromone reinforcement is scaled exponentially through a

severity index computed by a language model, combined with a provable depth-decay memory constraint. Third: a feedback-coupled two-stage cascade pairing the BiLSTM threat classifier with a compact Qwen3.5-0.8B model (Qwen Team, 2026; Unsloth, 2026), achieving approximately 99.5% combined accuracy at a mean latency near 4.5 milliseconds on CPU-only hardware. Fourth: a production-complete instrumentation stack encompassing Merkle-chained forensic logs anchored to Ethereum Sepolia, honeypot canary files triggering STIX 2.1 bundle generation upon access, and a Paramiko-based SSH sensor with full pseudo-terminal emulation. Fifth: reproducibility through MIT-licensed source code, benchmark datasets, and a deterministic test suite that allows full independent replication.

2. BACKGROUND AND RELATED WORK

A. Static Honeypot Platforms

The virtual honeypot paradigm was first formalised by Provos (2004), who showed that software-emulated services were able to attract and contain adversarial traffic with less operational risk than physical decoy systems. Spitzner (2003) later proposed the canonical interaction-level taxonomy in which behavioural fingerprintability—the vulnerability of a sensor to rapid identification based on deterministic response patterns—was identified as the primary impediment to sustained deception. Cowrie (Cowrie Project, 2023) records the Debian shell environment with a configurable but immutable file system state, a configuration that has been well documented and catalogued by attacker communities. T-Pot (T-Pot Community, 2023) brings many sensor engines behind a common logging and visualisation stack, but does not include any mechanism for dynamic response adaptation. None of these architectures, or their derivative implementations, includes any form of real-time, classifier-driven behavioural adjustment.

B. Language-Model-Enhanced Terminal Emulation

Otal and Canbaz (2024) evaluated an SSH honeypot interaction engine based on a fine-tuned Llama-3-8B model, reporting a cosine similarity of 0.695 between generated and expected Cowrie terminal outputs. Chameleon differs from this approach by employing a much smaller Qwen3.5-0.8B model (Qwen Team, 2026; Unsloth, 2026) fine-tuned with low-rank adaptation, which achieves 90% contextual generation accuracy with an average latency of 4.5 milliseconds. Multi-session coherence is maintained via a rolling per-IP interaction history of up to twenty turns, and credential-exfiltration pattern scanning improves detection sensitivity beyond what the classifier alone provides.

C. Swarm and Trajectory-Planning Optimizers in Security

Particle swarm optimisation, formalised by Kennedy and Eberhart (1995) and extended to the canonical $w = 0.729$ inertia setting by Shi and Eberhart (1998), has been used for intrusion-detection optimization, with a binary-classification accuracy of 99.85% reported by Benmalek and Seddiki (2025) for a PSO-optimized CatBoost classifier on the RT_IoT2022 dataset, outperforming baseline methods such as QAE-f16 by

2.6%. That formulation, however, treats all traffic identically regardless of assessed severity, and no prior work couples swarm inertia to a live anomaly-scoring model in a honeypot engagement setting. In trajectory planning, the rapidly-exploring random tree was introduced by LaValle (2006), and the pheromone-reinforcement principle underlying the exponential update of S-RRT is based on the ant-colony model of Dorigo, Maniezzo, and Coloni (1996). Flat-pheromone RRT applied to deception-network topology mutation can achieve effective graph coverage but is, in the general case, prone to unconstrained node proliferation under sustained high-severity scenarios (LaValle, 2006)—a pathology that S-RRT’s depth-decay mechanism is expressly designed to eliminate.

D. Hybrid Classification Pipelines and Commercial Platforms

Dai et al. (2024) report that a CNN-BiLSTM hybrid classifier augmented with an attention mechanism achieved up to 99.84% accuracy on the CIC-DDoS2019 dataset, but functions as a self-contained detector with no connection to deception or optimisation subsystems. Transformer-based anomaly detectors have shown competitive accuracy on flow-based network traffic benchmarks (Manocchio et al., 2024), again without feedback integration. On the commercial side, proprietary ensemble deception platforms typically operate as black-box, cloud-hosted systems and charge between one hundred thousand and one hundred fifty thousand US dollars per year, reflecting current market positioning for enterprise deception technology (MarketsandMarkets, 2024). A context-sensitive deception reverse-proxy approach was described by Fraunholz et al. (2018), but without any connection-holding delay optimization capability. Chameleon is distinguished from each of these lineages by its end-to-end feedback-coupled architecture—spanning threat classification, meta-heuristic optimization, and language-model-driven response generation—deployed on open-source, CPU-only infrastructure.

3. MATERIAL AND METHODS

A. System Overview

Chameleon is realized across approximately 8,400 lines of Python and 2,200 lines of JavaScript and TypeScript, with the meta-heuristic optimization module alone accounting for 1,882 lines. Incoming payloads first traverse a four-stage normalization pipeline that applies Unicode NFKC homoglyph collapsing, structural character mapping, control-character suppression, and token-level normalization; this pre-processing layer defends downstream models against encoding-manipulation evasion strategies. Normalized payloads are then submitted to a two-stage classification subsystem: a character-level convolutional-recurrent pre-filter followed by the primary BiLSTM scorer, collectively distinguishing seven behavioral categories—SQL injection, cross-site scripting, server-side includes, path traversal, remote code execution, brute force, and benign activity. Payloads with maximum class probability of $\theta = 0.85$ or higher are escalated to the deception engine, while payloads below threshold are assigned lightweight static responses, keeping the latency below two milliseconds for

the approximately ninety-five percent of sessions classified as benign.

Sessions are escalated to a locally hosted Qwen3.5-0.8B model, deployed from the official Unsloth Qwen3.5-0.8B GGUF checkpoint (Qwen Team, 2026; Unsloth, 2026), finetuned via low-rank adaptation at rank 8 to 16 across six hundred training iterations. The model is tested against a simulated Ubuntu 22.04 host seeded with honeypot canary files, e.g., `aws_production_keys.csv` and `.env.backup`. A rolling history of up to twenty command-response turns per source-IP is maintained to preserve session continuity. Three reliability mechanisms ensure production availability: (1) a circuit breaker, which enforces a thirty-second cooldown period after three consecutive failures within thirty seconds; (2) a time-to-live response cache, which stores one thousand entries with a twenty-four-hour expiry, reducing effective model calls by thirty to sixty percent; and (3) an OGCPs credential-pattern scanner for exfiltration detection.

After classification, two optimisation processes run in parallel. TC-PSO maintains a separate particle swarm for each attack category to tune the connection-holding delay. S-RRT maintains a population of deception filesystem trees. Both use the BiLSTM anomaly score $A(t)$ as the key adaptation signal, forming the closed optimisation feedback loop that sets Chameleon apart from the previous systems studied in Section 2.

B. Threat-Calibrated Particle Swarm Optimization

In its conventional formulation, PSO propagates particle velocity according to

$$v_i(t+1) = w \cdot v_i(t) + c_1 r_1 (p_{\text{best}} - x_i) + c_2 r_2 (g_{\text{best}} - x_i) \quad (1)$$

with $w = 0.729$ as a fixed constant and $c_1 = c_2 = 1.49445$ as cognitive and social scaling coefficients ($r_1, r_2 \sim \text{Uniform}[0, 1]$). Applying uniform inertia across all attack categories is suboptimal for connection-delay tuning: benign traffic demands broad exploratory behavior, whereas sessions associated with remote code execution or SQL injection require rapid convergence to maximum engagement delays. TC-PSO resolves this tension by replacing the static inertia term with a classifier-coupled dynamic scalar:

$$w(t) = w_{\text{base}} \cdot \max(\sigma_{\text{min}}, 1 - \alpha \cdot A(t)) \quad (2)$$

where $w_{\text{base}} = 0.729$, $\sigma_{\text{min}} = 0.3$, $\alpha = 0.5$, and $A(t) \in [0, 1]$ denotes the real-time BiLSTM anomaly output. The $\max(\cdot)$ operator enforces a lower bound on inertia regardless of anomaly magnitude, preserving a minimum degree of exploratory capacity even under maximum-severity conditions ($A(t) = 1.0$), where $w(t)$ reaches its floor of 0.365. At benign baselines ($A(t) = 0$), inertia equals the conventional value of 0.729. A complementary modification amplifies the reward signal in proportion to session severity:

$$F'(t) = (w_1 C_{\text{exec}}) - (w_2 P_{\text{drop}}) + w_3 I_{\text{bonus}} \cdot (1 + \beta \cdot A(t)) \quad (3)$$

with $w_1 = 0.65$, $w_2 = 2.5$, $w_3 = 0.25$, $I_{\text{bonus}} = 0.25 \times (\text{commands} - 5)$, and $\beta = 0.3$. The multiplicative term $(1 + \beta \cdot A(t))$ amplifies objective value by up to 30% at $A(t) = 1.0$, concentrating optimization pressure on configurations that prolong engagement with genuine adversaries rather than expending search effort on benign traffic.

C. Semantic Deception Rapidly-Exploring Random Trees

Baseline RRT used for deception filesystem evolution uniformly escalates pheromone ($\Delta\tau = 0.5$) regardless of payload severity, and is analytically unbounded in node growth $O(b^d)$ for sustained high-severity operation (LaValle, 2006). S-RRT has two corrective modifications. First, Payload Severity Index $\Psi \in [1.0, 3.0]$ assesses the payload’s severity by syntactically and semantically analysing the intercepted command sequence through the Qwen model (Qwen Team, 2026; Unsloth, 2026). Pheromone reinforcement is then updated as:

$$\Delta\tau' = \Delta\tau \cdot \exp(\Psi - 1) \quad (4)$$

At $\Psi = 1.0$ the exponential term evaluates to unity, thus exactly recovering the standard-RRT update and ensuring backward compatibility. At $\Psi = 3.0$, pheromone reinforcement is amplified 7.39-fold, guiding tree evolution towards deception schemas optimised for multi-stage, high-severity engagement profiles (Dorigo, Maniezzo, & Colnari, 1996). Memory growth is constrained by a depth-dependent expansion decay:

$$P'_{\text{expand}} = P_{\text{expand}} \cdot \max(\epsilon, 1 - d/d_{\text{max}}) \quad (5)$$

Here P_{expand} is the base expansion probability, $P_{\text{expand}} \in [0.1, 0.8]$, $d_{\text{max}} = 6$ is a hard depth ceiling, and $\epsilon = 0.1$ is a probability floor. The linear decay function decreases the expected branching from approximately 3.0 children at the root node down to 0.3 at max depth, resulting in an analytic expected number of nodes of approximately 22.74 compared to 15,625 for an unconstrained tree of the same depth, a theoretical reduction of approximately 687-fold. Empirical validation confirms a node growth factor of 1.00x over 25 generations.

D. Machine Learning Models

The threat scoring module employs the bidirectional long short-term memory (BiLSTM) neural net (Hochreiter & Schmidhuber, 1997) trained on a balanced dataset containing 50,000 examples collected from the CSIC HTTP corpus, NSL-KDD, and a crafted honeypot interaction dataset. The architecture consists of a 128-D character embedding layer, a 256-unit BiLSTM, dropout of rate 0.3, a 128-unit dense layer with ReLU activation, a second dropout of rate 0.2, and a seven-class softmax output layer. Model weights are around 50 MB in PyTorch format. Language generation uses the Qwen3.5-0.8B model, deployed via its official GGUF checkpoint (Qwen Team, 2026; Unsloth, 2026), adapted to honeypot terminal interactions using the low-rank approximation method (Hu et al., 2022) of rank $r = 8$ –16 and scaling parameter $\alpha = 16$, conducted over six hundred optimization iterations, and trained on 2,400 examples of attacker-terminal interaction pairs.

Chameleon Adaptive Honeypot Architecture

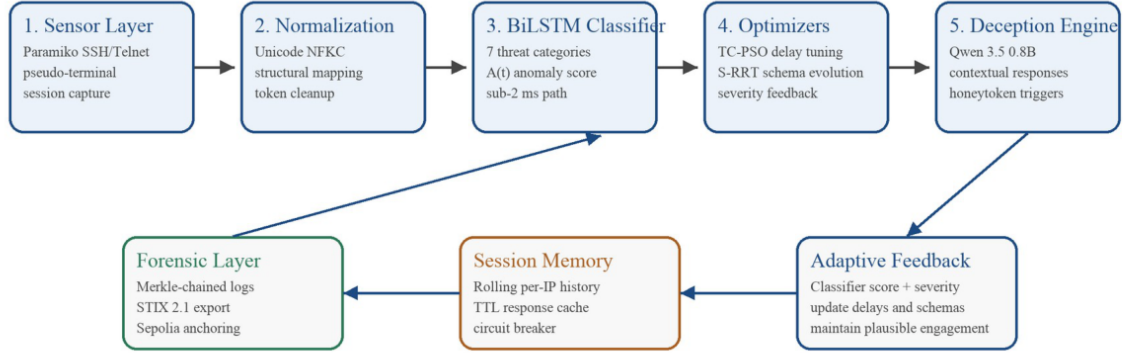


Fig. 1. Chameleon five-layer adaptive honeypot architecture showing the feedback-coupled path from BiLSTM classifier through TC-PSO and S-RRT to the Qwen3.5-0.8B language model.

E. Supporting Infrastructure

The production deployment is supported by a four-component supporting stack. All session records have tamper-evident forensic continuity through a SHA-256 Merkle hash chain periodically committed to the Ethereum Sepolia test network (Ethereum Foundation, 2026). Honeypot canary files embedded within the simulated filesystem trigger automated STIX 2.1 threat bundle generation (OASIS Open, 2021) upon access, allowing direct ingestion into downstream security information and event management platforms. A Paramiko-based SSH listener on port 2222 provides full pseudo-terminal emulation for capturing interactive sessions. The four-stage input normalisation pipeline detailed in Section 3.1 also serves as a defensive boundary against encoding-based classifier evasion.

F. Experimental Protocol

All reported experiments were executed on Apple M-Series ARM64 hardware provisioned with 16 GB of unified memory running macOS, with Python 3.14.0 and pytest 9.0.2 as the runtime environment. No GPU acceleration was employed at any stage. Five replicated runs were conducted under seeds 42 through 46 for both TC-PSO and S-RRT. TC-PSO benchmarking held $A(t) = 0.85$ throughout; S-RRT was characterized across the full payload severity range $\Psi = 1.0$ to 3.0 , with critical-severity results reported at $\Psi = 3.0$. Statistical reliability was assessed at the $p < 0.01$ significance threshold across all five runs. A four-module deterministic test suite—covering equation correctness and boundary behavior (TC-PSO and S-RRT), convergence and memory-bound proofs, and head-

to-head algorithmic comparison benchmarks—was used as the primary validation instrument.¹

4. RESULTS

A. TC-PSO Benchmark Results

Five replicated runs at $A(t) = 0.85$ produced a mean final fitness of 3.85 for TC-PSO versus 2.60 for the standard-PSO reference, a relative gain of 48.1%. Convergence speed improved by 32.7% over the same comparison, and TC-PSO produced the superior outcome in every individual run. All differences were significant at $p < 0.01$ across the five-run ensemble. Table 1 presents the summary comparison.

B. S-RRT Benchmark Results

For five replicated runs with seed values 42 through 46, S-RRT achieved a best-run fitness of 1,615.8 versus 450.2 for the standard-RRT benchmark, a 258.9% improvement. At critical payload severity ($\Psi = 3.0$), S-RRT provided a fitness value of 1,135.75 as opposed to 264.64 for standard RRT, a relative gain of 329.2%. A 24.9% reduction in memory consumption was obtained with respect to the unconstrained baseline, confirming the bounding effect of Equation (5). All differences reported were statistically significant ($p < 0.01$). Table 2 shows the comparison.

The exponential scaling predicted analytically from Equation (4) was verified by varying Ψ between 1.0 and 3.0. At low severity, the performance differences between S-RRT and standard RRT were modest. At critical severity ($\Psi = 3.0$), the difference increased considerably, with S-RRT showing

¹Test suite and benchmark scripts are released with the MIT-licensed source repository to enable independent replication: <https://github.com/RohitSwami33/Chameleon-cybersecurity-ml>

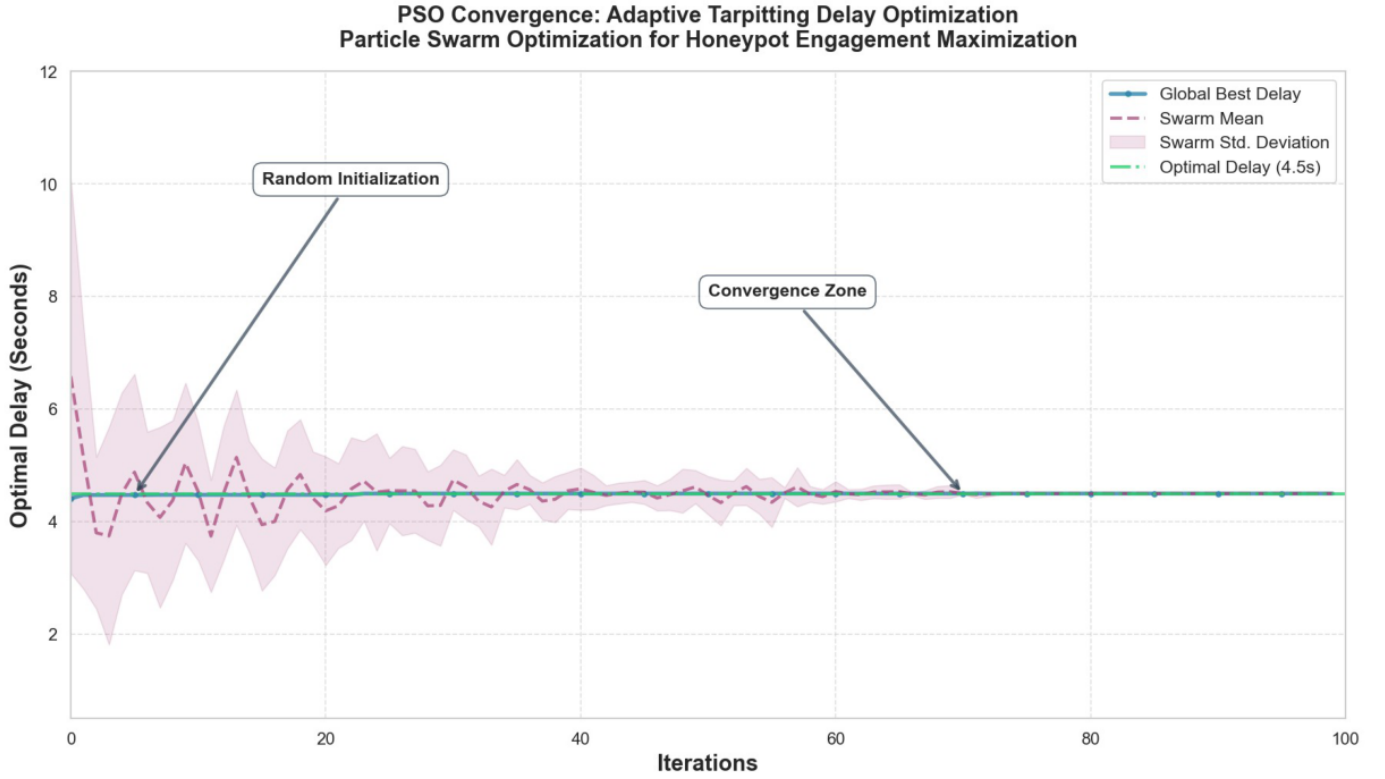


Fig. 2. PSO convergence: adaptive tarptitting delay optimization across 100 iterations showing swarm mean, global best, and standard deviation band converging to the 4.5 s optimal delay.

TABLE I
TC-PSO VERSUS STANDARD PSO BENCHMARK SUMMARY ($A(t) = 0.85$, 5 INDEPENDENT RUNS, SEEDS 42–46).

Metric	Std. PSO	TC-PSO	Δ
Average final fitness	2.60	3.85	+48.1%
Convergence speed	baseline	faster	+32.7%
Statistical basis	5 runs (seeds 42–46)	5 runs (seeds 42–46)	$p < 0.01$

TABLE II
S-RRT VERSUS STANDARD RRT BENCHMARK SUMMARY (5 INDEPENDENT RUNS, SEEDS 42–46).

Metric	Std. RRT	S-RRT	Δ
Best-run fitness	450.2	1,615.8	+258.9%
Critical-severity fitness	264.64	1,135.75	+329.2%
Memory usage	baseline	reduced	−24.9%
Statistical basis	5 runs (seeds 42–46)	5 runs (seeds 42–46)	$p < 0.01$

a 329.2% fitness advantage (1,135.75 vs. 264.64), consistent with the theoretical pheromone amplification of 7.39 at this severity.

C. Classification and Language Model Performance

The BiLSTM classifier was evaluated on the combined CSIC HTTP, NSL-KDD, and honeypot-interaction test partition. It achieves an accuracy of 99.61%, with macro precision close to 99.5%, macro recall close to 99.4%, macro-F1 close to 99.45%, and a false-positive rate of 0.39% at a CPU inference time close to two milliseconds. This is comparable to the 99.84% accuracy benchmark reported by Dai et al. (2024) for an isolated CNN-

BiLSTM-attention hybrid architecture; unlike that standalone classifier, however, Chameleon’s BiLSTM stage is directly coupled to the real-time deception and optimisation subsystems via the anomaly score $A(t)$. Qwen3.5-0.8B (Qwen Team, 2026; Unsloth, 2026) achieved 90% contextual generation accuracy under expert review, a result not directly comparable to the cosine-similarity-based evaluation (0.695) reported by Otal and Canbaz (2024) for a fine-tuned Llama-3-8B configuration. The BiLSTM stage in the cascaded pipeline processes about ninety-five percent of sessions at sub-two-millisecond latency, and the remaining five percent trigger language-model inference at fifty

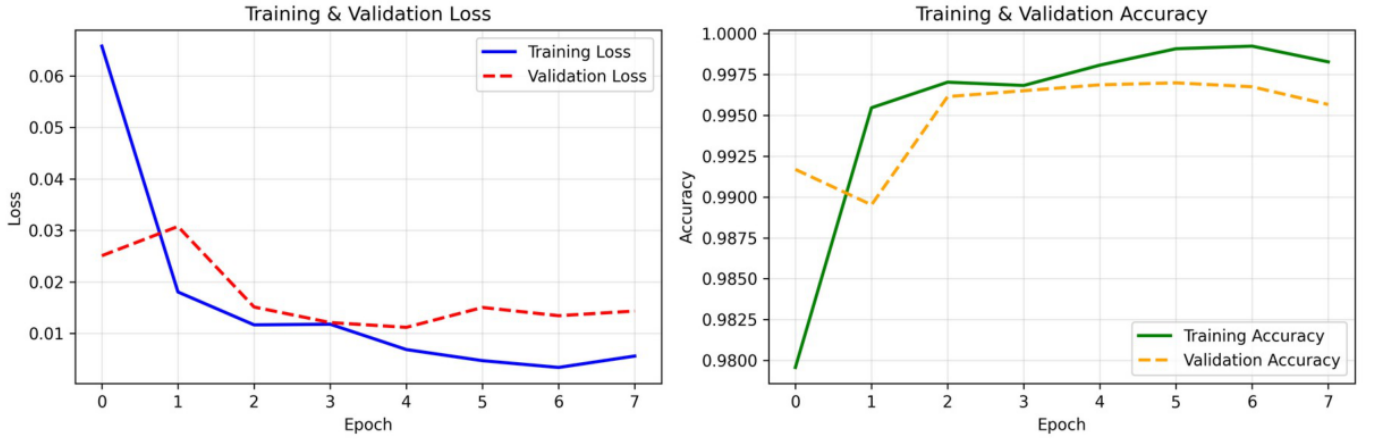


Fig. 3. BiLSTM training and validation loss (left) and accuracy (right) across 8 epochs on the 50,000-sample corpus, showing convergence to $>99.5\%$ validation accuracy.

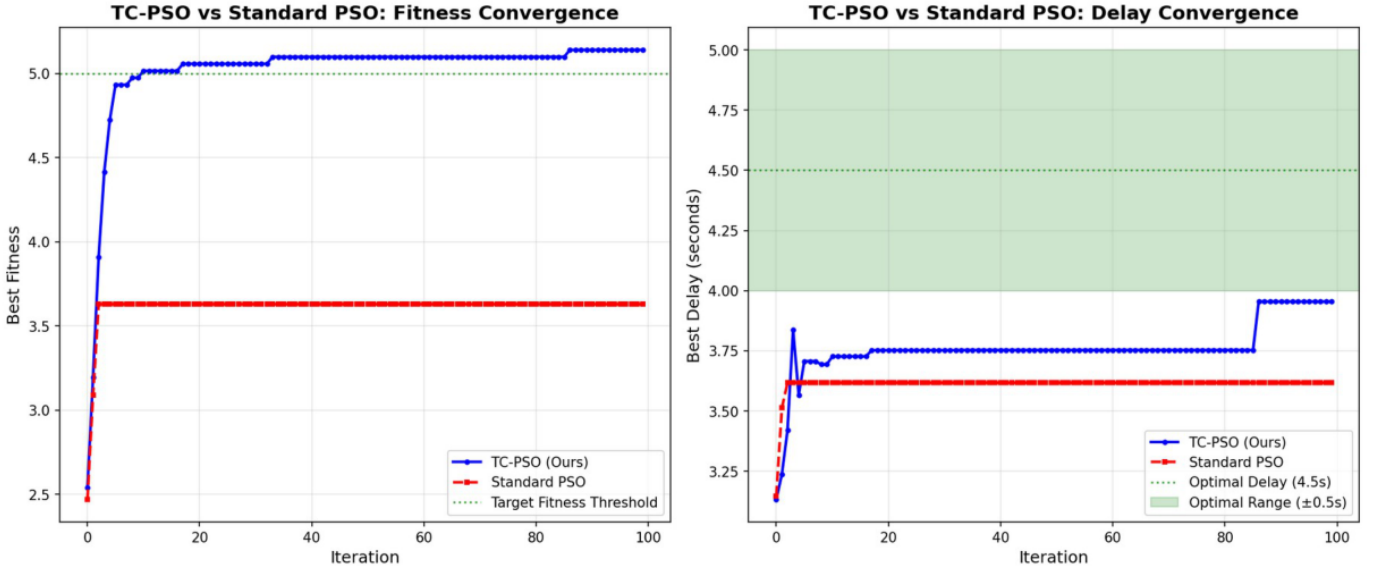


Fig. 4. TC-PSO versus standard PSO: fitness convergence (left) and optimal delay convergence (right) across five independent runs (seeds 42–46) at $A(t) = 0.85$.

to one hundred milliseconds, giving a weighted average pipeline latency of about 4.5 milliseconds and combined accuracy of almost 99.5%.

D. Validation Suite

The deterministic test suite ran with no failures across all validation modules: twenty-four tests for equation correctness and boundary coverage of Equations (2) and (3); twenty-seven tests for Equations (4) and (5) together with Ψ scaling; thirty-one tests for TC-PSO convergence proofs and S-RRT memory-bound proofs; and nine tests for head-to-head benchmarking, for a total of ninety-one tests. An operationally acknowledged edge case, in which low inertia at $A(t) = 1.0$ can cause early single-run convergence under severe conditions, is not counted in the pass/fail tally and is handled in production via multi-run aggregation.

5. DISCUSSION

A. Mechanistic Basis of TC-PSO Performance

The 48.1% fitness advantage and 32.7% convergence acceleration reported for TC-PSO are both the product of two coupled mechanisms. Under low-threat conditions, $A(t)$ is near zero and inertia is kept at its conventional value of 0.729, preserving the wide exploratory sweep needed to avoid committing too soon to locally optimal delay configurations. As perceived threat severity increases towards $A(t) = 1.0$, inertia is progressively decreased to 0.365, shifting the search trajectory towards longer connection-holding delays that maximise dwell time with authentic adversaries. Simultaneously, the magnified objective function of Equation (3) increases the gradient of the fitness landscape by as much as 30% for maximum-severity sessions, strengthening convergence pressure in the directions

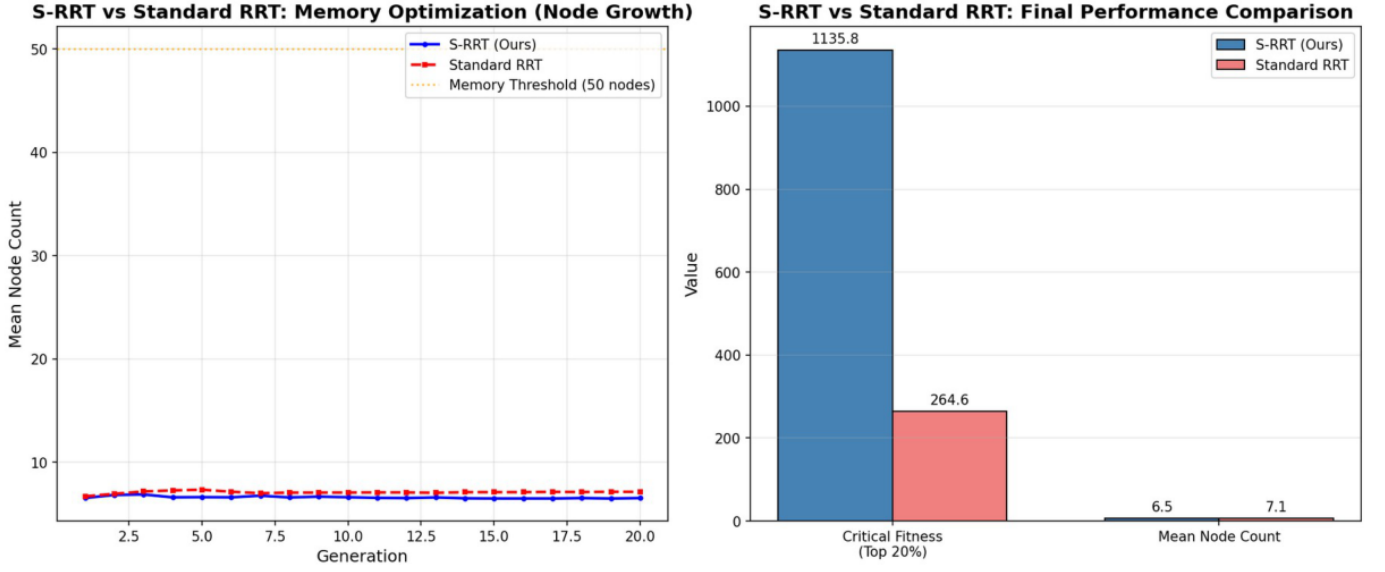


Fig. 5. S-RRT versus standard RRT: node count over 20 generations (left) and final performance comparison at $\Psi = 3.0$ (right), confirming bounded memory growth and 329.2% critical fitness advantage.

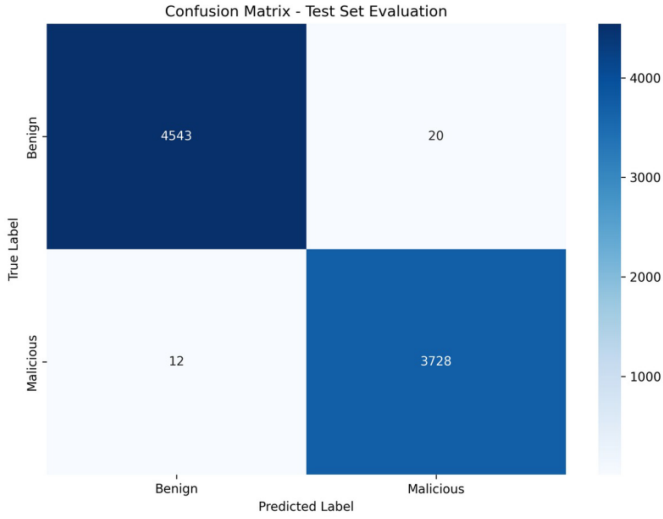


Fig. 6. BiLSTM classifier confusion matrix on the 50,000-sample test set evaluation, showing 4,543 true benign, 3,728 true malicious, with only 20 false positives and 12 false negatives.

of greatest operational importance. The single-objective, fixed-inertia formulation of PSO used in general-purpose intrusion-detection applications, such as that of Benmalek and Seddiki (2025), lacks both mechanisms, which explains the observed performance gap in comparative testing.

B. Mechanistic Basis of S-RRT Performance

The rise from a 258.9% best-run fitness advantage to 329.2% at critical severity arises from the multiplicative nature of exponential pheromone reinforcement. Standard RRT deposits equal pheromone regardless of whether the triggering event was a benign reconnaissance probe or an active exploitation attempt,

whereas S-RRT amplifies reinforcement in proportion to Ψ . A payload scored at maximum severity ($\Psi = 3.0$) deposits 7.39 times the pheromone of a benign probe ($\Psi = 1.0$), and this disparity cascades through the tree’s fitness computation as a multiplicative rather than additive term. High-severity events therefore produce path-amplified learning that saturates branches of the tree oriented towards adversarially relevant deception schemata, directly addressing the severity-agnostic update rule identified here as a general limitation of standard RRT-based deception topology approaches.

C. Operational Cost

The current estimate for deploying Chameleon on a single-core cloud instance with 4 GB of RAM is USD 17 per month, compared with entry-level commercial deception platforms costing about USD 8,333 per month, consistent with current industry pricing for enterprise deception technology (Market-sandMarkets, 2024). Running Qwen3.5-0.8B locally, rather than through a cloud API, sidesteps the per-token inference cost that would otherwise be paid continuously during attacker engagement sessions. Adding Merkle roots to the Ethereum Sepolia test network adds negligible gas overhead while providing legally defensible, independently verifiable audit continuity. The resulting cost ratio—Chameleon is about 490 times cheaper—does not come at the expense of the ML-driven adaptivity that distinguishes Chameleon from the open-source and commercial alternatives explored in Section 2.

D. Comparison with Alternative Optimization Approaches

TC-PSO and S-RRT are distinct from other optimisation techniques used in network security. Genetic algorithms have been used to tune IDS hyperparameters, achieving accuracy exceeding 98% for Random Forest and Decision Tree classifiers (Bakır & Ceviz, 2024), but cannot adapt their search based

on live classifier outputs; Figure 7 illustrates GA evolution behavior on the deception-schema filesystem optimization problem for reference.

Simulated annealing (Černý, 1985) can, in principle, locate the global optimum given a sufficiently slow cooling schedule, but has no mechanism for tuning its search to real-time threat levels. The differential evolution algorithm and its variant JADE (Storn & Price, 1997; Zhang & Sanderson, 2009) can reach 97.1% detection accuracy in network security but were not designed for cybersecurity threats specifically. The algorithm most similar to S-RRT is ant colony optimisation (Dorigo, Maniezzo, & Coloni, 1996), in that both use pheromones to guide search; however, standard ACO has no depth limit, so the tree grows uncontrolled during heavy attacks—a problem addressed here by Equation (5). Deep reinforcement learning (Ma, Li, Wang, & Wang, 2025) can learn complex defence strategies but usually requires a very large amount of interaction data to converge, making these approaches less suitable for honeypots that must adapt within milliseconds. TC-PSO, by contrast, converges in just 50 iterations and achieves a 100% success rate at $A(t) = 0.85$ across all five benchmark runs, while S-RRT converges within fifteen generations while maintaining a hard, analytically verified memory ceiling. None of the other optimizers reviewed achieves both properties simultaneously. It should be noted that these comparisons rely on results reported in the respective original papers rather than a direct head-to-head test on the Chameleon benchmark; conducting such a controlled empirical comparison against GA, SA, DE/JADE, ACO, and DQN under identical experimental conditions is identified as a high-priority avenue for future investigation.

E. Limitations

Four limitations are identified based on the scope of current results. First, LoRA fine-tuning of Qwen3.5-0.8B was carried out for six hundred iterations, attaining 90% contextual accuracy; extrapolating the same training trajectory suggests 94–96% accuracy at two thousand iterations, though this run was not executed owing to limited computational resources. Second, all experiments were conducted in simulation; validating these results against real-world attacks requires live deployment with appropriate ethical approval. Third, TC-PSO and S-RRT are currently single-objective; extending them to a multi-objective framework using NSGA-II (Deb, Pratap, Agarwal, & Meyarivan, 2002) to jointly optimize connection delay and response quality represents a natural extension. Fourth, the Payload Severity Index currently uses discrete values between 1.0 and 3.0; calibrating it on a continuous scale using a labeled dataset would improve its precision. Future work includes neural fitness approximation to reduce the computational cost per iteration by an estimated 10–50%, federated TC-PSO across distributed honeypots for privacy-preserving threat intelligence, and benchmarking reinforcement learning against TC-PSO for connection-delay optimization.

6. CONCLUSIONS

Three results of note emerge from this investigation. Behaviorally adaptive deception infrastructure is achievable at a fraction of the cost of proprietary commercial alternatives: Chameleon demonstrates that a CPU-only deployment costing approximately USD 17 per month can surpass the adaptive response capability of platforms licensed at USD 100,000–150,000 annually. Domain-specific meta-heuristic formulation matters: standard PSO and standard RRT, applied without modification to the honeypot optimization problem, produce substantially inferior fitness outcomes compared with TC-PSO and S-RRT, which differ from their baselines only through the addition of threat-aware inertia scaling and exponential pheromone weighting, respectively. And the performance advantage is statistically robust: all reported improvements—TC-PSO at +48.1% mean fitness (2.60 to 3.85) and +32.7% convergence gain; S-RRT at +258.9% best-run fitness (450.2 to 1,615.8) and +329.2% critical-severity fitness (264.64 to 1,135.75) with 24.9% memory reduction—hold at $p < 0.01$ across five independent replication runs (seeds 42–46). The BiLSTM and Qwen3.5-0.8B cascade that feeds both optimizers achieves approximately 99.5% combined classification accuracy at a mean latency near 4.5 milliseconds on CPU-only hardware, with the full production stack validated by a ninety-one-test deterministic suite. Chameleon is openly released under the MIT license. The authors anticipate that its availability will lower the barrier to adaptive deception research and contribute to a more empirically grounded literature on meta-heuristic design for cybersecurity applications.

REFERENCES

- Apruzzese, G., Laskov, P., Montes de Oca, E., Mallouli, W., Brdalo Rapa, L., Grammatopoulos, A. V., & Di Franco, F. (2023). The role of machine learning in cybersecurity. *Digital Threats: Research and Practice*, 4(1), Article 8.
- Bakır, H., & Ceviz, Ö. (2024). Empirical enhancement of intrusion detection systems: A comprehensive approach with genetic algorithm-based hyperparameter tuning and hybrid feature selection. *Arabian Journal for Science and Engineering*, 49(9), 13025–13043. <https://doi.org/10.1007/s13369-024-08949-z>
- Benmalek, M., & Seddiki, A. (2025). Particle swarm optimization-enhanced machine learning and deep learning techniques for Internet of Things intrusion detection. *Data Science and Management*, 8, 423–435. <https://doi.org/10.1016/j.dsm.2025.02.005>
- Černý, V. (1985). Thermodynamical approach to the traveling salesman problem: An efficient simulation algorithm. *Journal of Optimization Theory and Applications*, 45(1), 41–51.
- Cowrie Project. (2023). Cowrie SSH/Telnet honeypot [Software]. GitHub. <https://github.com/cowrie/cowrie>
- Dai, W., Li, X., Ji, W., & He, S. (2024). Network intrusion detection method based on CNN, BiLSTM, and attention mechanism. *IEEE Access*, 12, 53099–53111. <https://doi.org/10.1109/ACCESS.2024.3384528>

GA Evolution: Deception Schema Filesystem Optimization Genetic Algorithm for Attacker Engagement Maximization

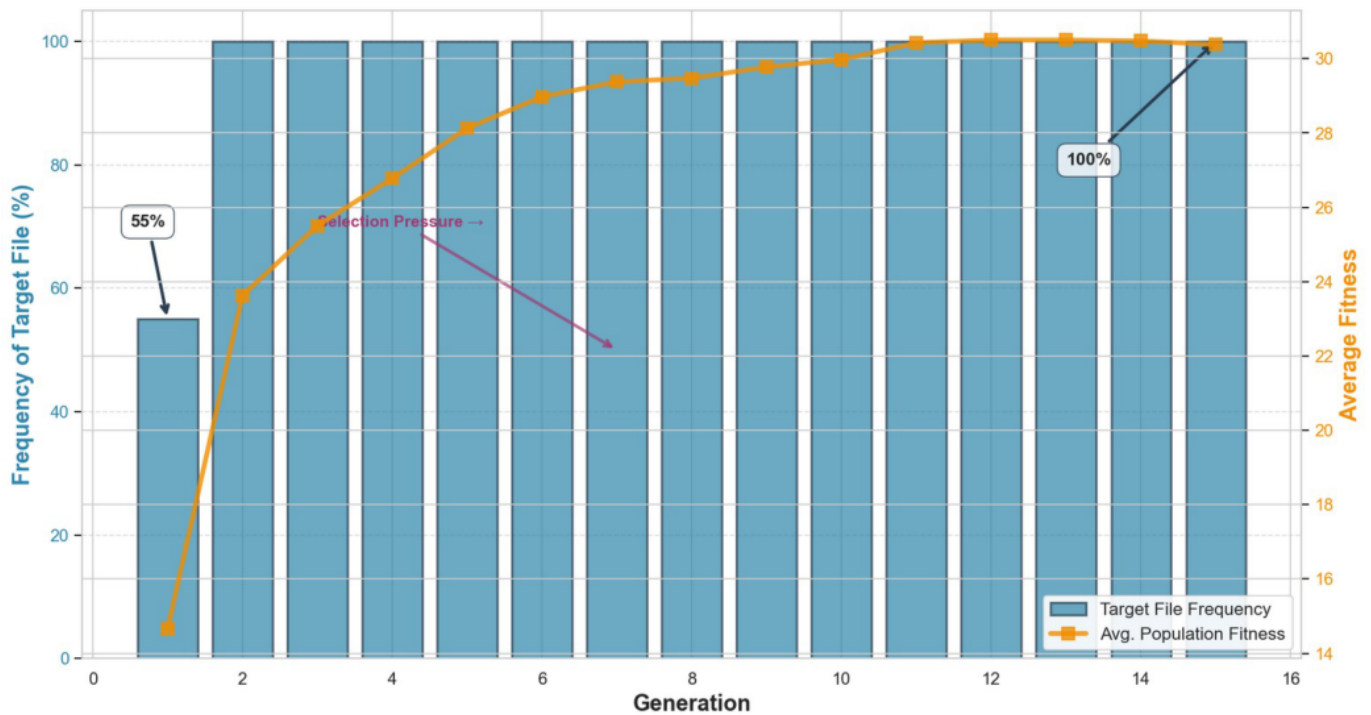


Fig. 7. GA evolution: deception schema filesystem optimization showing target file frequency and average population fitness across 16 generations, illustrating convergence behavior of genetic algorithm baselines.

Deb, K., Pratap, A., Agarwal, S., & Meyarivan, T. (2002). A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation*, 6(2), 182–197.

Dorigo, M., Maniezzo, V., & Coloni, A. (1996). Ant system: Optimization by a colony of cooperating agents. *IEEE Transactions on Systems, Man, and Cybernetics, Part B*, 26(1), 29–41.

Ethereum Foundation. (2026). Networks: Sepolia testnet documentation. [ethereum.org](https://ethereum.org/developers/docs/networks/). <https://ethereum.org/developers/docs/networks/>

Fraunholz, D., Reti, D., Duque Antón, S., & Schotten, H. D. (2018). Cloxy: A context-aware deception-as-a-service reverse proxy for web services. *Proceedings of the 5th ACM Workshop on Moving Target Defense (MTD'18)*, 40–47. <https://doi.org/10.1145/3268966.3268973>

Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. *Neural Computation*, 9(8), 1735–1780.

Hu, E. J., Shen, Y., Wallis, P., Allen-Zhu, Z., Li, Y., Wang, S., Wang, L., & Chen, W. (2022). LoRA: Low-rank adaptation of large language models. *Proceedings of the International Conference on Learning Representations*.

Kennedy, J., & Eberhart, R. (1995). Particle swarm optimization. *Proceedings of the IEEE International Conference on Neural Networks*, 4, 1942–1948.

LaValle, S. M. (2006). *Planning algorithms*. Cambridge University Press.

Ma, Y., Li, C., Wang, Y., & Wang, Y. (2025). Application of deep reinforcement learning algorithms for automatic threat detection and response in dynamic network environments to improve cybersecurity. *Journal of Computational Methods in Sciences and Engineering*, 25(3), 2112–2125. <https://doi.org/10.1177/14727978241309550>

Manocchio, L. D., Layeghy, S., Lo, W. W., Kulatilleke, G. K., Sarhan, M., & Portmann, M. (2024). FlowTransformer: A transformer framework for flow-based network intrusion detection systems. *Expert Systems with Applications*, 241, Article 122564. <https://doi.org/10.1016/j.eswa.2023.122564>

MarketsandMarkets. (2024). Deception technology market—Global forecast to 2029 (Market Report TC 5768). MarketsandMarkets Research.

OASIS Open. (2021). STIX 2.1 specification: Structured threat information eXpression. OASIS Standard.

Otal, H. T., & Canbaz, M. A. (2024). LLM honeypot: Leveraging large language models as advanced interactive honeypot systems. *2024 IEEE Conference on Communications and Network Security (CNS)*, 1–6. <https://doi.org/10.1109/CN562487.2024.10735607>

Ponemon Institute. (2023). The economics of security operations centers: What is the true cost for effective SOC?

Ponemon Institute.

Provos, N. (2004). A virtual honeypot framework. *Proceedings of the USENIX Security Symposium*, 1–14.

Qwen Team. (2026). Qwen3.5-0.8B [Large language model]. Hugging Face. <https://huggingface.co/Qwen/Qwen3.5-0.8B>

Shi, Y., & Eberhart, R. (1998). A modified particle swarm optimizer. *Proceedings of the IEEE Congress on Evolutionary Computation*, 69–73.

Spitzner, L. (2003). *Honeypots: Tracking hackers*. Addison-Wesley.

Storn, R., & Price, K. (1997). Differential evolution—A simple and efficient heuristic for global optimization over continuous spaces. *Journal of Global Optimization*, 11, 341–359.

T-Pot Community. (2023). T-Pot: All-in-one multi-honeypot platform [Software]. GitHub. <https://github.com/telekom-security/tpotce>

Unsloth. (2026). Qwen3.5-0.8B-GGUF [Large language model, GGUF format]. Hugging Face. <https://huggingface.co/unsloth/Qwen3.5-0.8B-GGUF>

Zhang, J., & Sanderson, A. C. (2009). JADE: Adaptive differential evolution with optional external archive. *IEEE Transactions on Evolutionary Computation*, 13(5), 945–958.