Electronics and Electrical Engineering

AIMS Electronics and Electrical Engineering, 10(2): 314–333.
DOI: 10.3934/electreng.2026013
Received: 07 January 2026
Revised: 15 March 2026
Accepted: 30 March 2026
Published: 10 April 2026

https://www.aimspress.com/journal/ElectrEng

Research article

Evo-TTP: Generative and robust prediction of novel cyber threat tactics using adversarial fine-tuning of large language models

Dilkhaz Mohammed* and Shahram Jamali

Department of Computer Engineering, University of Mohaghegh Ardabili, Ardabil, Iran

*** Correspondence:** Email: dilkhaz.mohammed@uma.ac.ir; Tel: +9647508111640.

Abstract: The asymmetrical nature of the modern cyber threat landscape allows advanced persistent threats (APTs) to innovate tactics at a significantly faster rate than defensive frameworks can document them. While the MITRE ATT&CK® framework provides a standardized taxonomy of known behaviors, it essentially functions as a retrospective database — a "dictionary of the past" which fails to anticipate future "zero-day" tactics, techniques, and procedures (TTPs). This paper introduces Evo-TTP, a comprehensive framework for the predictive generation of novel and robust tactics, techniques, and procedures via big data mining and adversarial learning. By leveraging the massive structured data from the MITRE ATT&CK® Enterprise Matrix v18.0, Evo-TTP treats the prediction of future threats as a high-dimensional pattern completion problem. Our methodology addresses two primary failures in current generative AI applications to big data: mode collapse, which refers to hallucinating biologically or technically impossible scenarios, and algorithmic brittleness, which is characterized by its susceptibility to adversarial perturbations. We employ a tripartite approach: (1) applying semantic pattern mining on the v18.0 dataset to create a baseline knowledge graph that reveals hidden correlations; (2) utilizing synthetic novelty expansion with a teacher-student architecture, using Llama-3.1-405B as the teacher and Llama-3.1-8B as the student model, to overcome data scarcity; and (3) conducting training in adversarial group relative policy optimization (GRPO). This training regime maximizes a composite reward function by balancing novelty, technical feasibility, and resilience against adversarial noise. Validated against the 2025 benchmarks and vetted according to SafeGen-X principles, Evo-TTP demonstrates a 23.1% increase in utility and an 18.2% improvement in robustness to adversarial attacks when compared with standard fine-tuning methods. This research positions generative AI not only as a text processor but also as a critical instrument in big data for uncovering the hidden evolutionary mechanics of cyberwarfare.

Keywords: large language models (LLMs); cybersecurity; MITRE ATT&CK v18.0; adversarial fine-tuning; group relative policy optimization (GRPO); big data mining; predictive threat intelligence; hidden pattern discovery

1. Introduction

The contemporary cyber threat landscape is defined by constant evolution. State-sponsored actors, cybercrime syndicates, and hacktivists operate with an agility that renders traditional signature-based defenses increasingly obsolete. Historically, defenders have been trapped in a "cycle of reaction," where they patch vulnerabilities and update detection rules only after a potential threat has been observed in real-world conditions. This reactive posture is fundamentally tied to the retrospective nature of threat intelligence. As noted by Rani et al. [1], effective defense requires "decoding shadows:" inferring the unseen elements based on known telemetry [1]. This approach allows defenders to anticipate the adversary's next pivot.

However, extrapolating future tactics, techniques, and procedures (TTPs) is a resource-intensive challenge in big data. The sheer volume of telemetry, combined with the complexity of modern cloud and hybrid environments, creates a vast search space of potential attack permutations. large language models (LLMs) have emerged as potential force multipliers in the cybersecurity domain, as they are capable of analyzing massive textual corpora to identify attack chains. Ali and Ghanem [2] argue that LLMs represent the next generation of cybersecurity defense by transforming data detection processes into actionable intelligence [2].

Despite this promise, the direct application of LLMs in predictive cyber threat intelligence (CTI) faces significant research challenges. First, standard models suffer from stochastic parroting, regurgitating known attacks rather than mining hidden patterns or identifying structural gaps in defense architectures [3]. Second, and perhaps more critically, recent findings by Mezzi et al. [4] illustrate the significant vulnerability of LLMs in security applications [4]. Minor adversarial perturbations whether intentional prompt injections or unintentional noise can lead to catastrophic failures in reasoning or safety guardrails. Consequently, these insights are unreliable for automated defense.

This paper proposes Evo-TTP, a framework designed to bridge the gap between static knowledge and the dynamic prediction of cyber threats. We hypothesize that by mining the structured logic of the MITRE ATT&CK v18.0 dataset, a widely used knowledge base for cyber threat intelligence and subjecting a generative model to rigorous testing, we can enhance predictive accuracy. Through adversarial fine-tuning, we enable the model to learn the underlying causal logic of cyberattacks rather than merely their surface-level descriptions. We present a "big data solution" that integrates semantic graph mining with adversarial group relative policy optimization (GRPO), a technique designed to enhance decision making under adversarial conditions to produce innovative insights resilient against various threats.

2. Background and related work

2.1. MITRE ATT&CK framework and semantic mining

The MITRE ATT&CK® framework serves as a global ontology for adversary behavior. Version

18.0 encapsulates a graph of tactics, techniques, and software. Traditionally used as a static lookup table, recent research has shifted toward semantic mining. For example, Graham [5] utilized semantic network analysis to identify the centrality and influence of various techniques, demonstrating the existence of hidden clusters [5]. Stein [6] demonstrated that "proactive cybersecurity" relies on mining large cybersecurity datasets for predictive indicators rather than reactive indicators of compromise (IoCs) [6]. Similarly, Ouaissa et al. [7] applied comparable frameworks to smart city environments, demonstrating that high-dimensional data analysis can reveal threat vectors invisible to manual review [7].

2.2. LLM-based cyber threat intelligence extraction

The application of LLMs to CTI has advanced significantly in recent years. Recent comprehensive surveys provide structured taxonomies for this field. Alam et al. [8] introduced CTIBench, a systematic benchmark for evaluating LLM applications in cybersecurity across multiple dimensions, including knowledge extraction, threat classification, and attack scenario synthesis [8]. Büchel et al. [9], in their systems of knowledge (SoK) paper on automated TTP extraction, examined LLM-based CTI extraction methods and demonstrated that fine-tuned models achieve higher precision than zero-shot approaches on TTP extraction tasks [9]. SynthCTI (2025) pioneered LLM-driven synthetic CTI generation, demonstrating that transformer architectures can synthesize coherent attack scenarios when trained on structured threat databases, thereby enhancing MITRE ATT&CK classification [10]. Ong et al. [11] explored LLM-driven threat synthesis for security evaluation using red teaming approaches and highlighted the challenges of generating adversarial content and evaluating model robustness [11]. Das et al. [12] provided a comprehensive survey of large language models (LLMs) in security research, identifying key challenges such as hallucination, adversarial brittleness, and evaluation methodologies [12]. Building on these foundational works, we categorize the existing approaches into three primary paradigms:

- Entity extraction and structuring: Lekssays et al. [13] and Cuong Nguyen et al. [14] utilized LLMs to parse unstructured big data reports into structured STIX entities [13,14]. TTPHunter [15] employed transformer-based models specifically for extracting TTPs from threat reports, achieving an 85% F1 score on benchmark datasets [15]. Similarly, Loumachi et al. [16] advanced cyber incident timeline analysis through retrieval-augmented generation, demonstrating improved accuracy in temporal event extraction [16].
- Knowledge graph construction: AttacKG+ [17] constructed comprehensive attack knowledge graphs from CTI reports using large language models (LLMs) with prompting-based approaches and zero-shot learning techniques [18]. Ma et al. [19] proposed APT-KG2QA, an intelligent fine-tuning strategy utilizing APT knowledge graphs to enhance LLM performance in threat intelligence question answering [19]. Loevenich et al. [20] automated cyber threat intelligence and attack chain generation using cybersecurity knowledge graphs combined with LLMs [20].
- Classification and detection: SecureBERT [21] introduced domain-specific pre-training for cybersecurity text classification, achieving state-of-the-art results on malware classification tasks with F1 = 0.73 [21]. Zhang et al. [22] researched TTP data augmentation methods based on the ATT&CK framework, improving classification accuracy through synthetic data generation [22].
- Generative Approaches: Ren et al. [23] proposed automated tactics planning for cyber-attack and defense based on LLM agents [23]. However, as ElZemity et al. [24] demonstrated, models

fine-tuned on 'pseudo-malicious' data often generate attacks that are technically infeasible [24]. Huang et al. [25] identified that this hallucinated threat intel closely resemble legitimate styles, which complicates validation [25].

While existing methods excel at extracting and classifying known threats, they fundamentally operate retrospectively, processing documented attacks rather than predicting novel ones. Evo-TTP addresses this gap by introducing predictive generation through structural hole analysis, a capability absent in current LLM-based CTI approaches.

2.3. Adversarial robustness in security-critical LLMs

A critical research issue in Big Data AI is brittleness. Angioni et al. [26] argued that models in security domains require "robustness-congruent" training to effectively handle adversarial noise [26]. To address this issue, Wu et al. [27] proposed "adversarial low-rank adaptation" (ALoRA) as a method for efficient and robust training [27]. Building on the work of Ji et al. [28], bi-GRPO (bidirectional group relative policy optimization), an extension of GRPO (group relative policy optimization), was introduced to prevent backdoor injections [28]. Zhou et al. [29] advanced the defense against adversarial examples by modeling adversarial noise [29]. Li et al. [30] proposed an effective adversarial defense framework from a robust feature perspective [30]. SecFlow [31] introduced an agentic LLM-based framework for modular cyberattack analysis and explainability, building on early approaches to adversarial fine-tuning for prompt injection defense [32], which provides important context for our robustness considerations [31]. Belcastro et al. [33] demonstrated the relevance of knowledge-graph representations as a foundation for explainable security reasoning [33]. Additionally, Belcastro et al. [34] addressed the infrastructural dimension required to support reliable large-scale training and evaluation [34]. Mamalakis et al. [35] contributed to the growing methodological focus on stable interpretability as a prerequisite for trustworthy model behavior under distribution shift and adversarial pressure [35].

Evo-TTP synthesizes these approaches. It applies GRPO specifically to the domain of threat logic, ensuring that the insights generated from this application are reliable.

3. Theoretical foundation: Structural hole analysis for hidden attack pattern detection

This section provides a theoretical foundation that explains why structural hole analysis in semantic graphs enables the detection of hidden attack patterns. We present both the mathematical framework and practical demonstrations of structural hole analysis.

3.1. Graph-theoretic foundation

The MITRE ATT&CK framework can be formally represented as a knowledge graph, G = (V, E, W), as shown in Table 1, where:

- $V = \{v_1, v_2, \ldots, v_n\}$ represents the set of nodes (tactics, techniques, procedures, malware, groups)
- $E \subseteq V \times V$ represents the set of edges (relationships: "uses", "mitigates", "sub-technique-of")
- $W: E \rightarrow \mathbb{R}$ represents edge weights indicating relationship strength

Definition 1 (Semantic embedding): Each node v_i is mapped to a high-dimensional vector space through a semantic embedding function f: V → $\mathbb{R}^d$, where d is the embedding dimension. We employ sentence-BERT to generate embeddings:

$$e_i = f(v_i) = \text{SBERT}\big(\text{description}(v_i)\big)$$

Definition 2 (Semantic similarity): The semantic similarity between two nodes is computed using cosine similarity:

$$\text{sim}\big(v_i, v_j\big) = \frac{e_i \cdot e_j}{\| e_i \| \cdot \| e_j \|}$$

Definition 3 (Structural hole): A structural hole exists between nodes v_i and v_j if and only if:

$$\text{sim}\big(v_i, v_j\big) > \tau \wedge \big(v_i, v_j\big) \notin E$$

where τ is the similarity threshold (we use τ = 0.75, selected via grid search; see Section 4.2).

Table 1. Notation summary.

Symbol	Definition
G	Knowledge graph
V	Set of nodes (tactics, techniques, procedures)
E	Set of edges (relationships)
e_i	Semantic embedding of node v_i
Sim (v_i, v_j)	Cosine similarity between embeddings
T	Similarity threshold for structural hole detection
H	Set of hidden patterns (structural holes)
D_syn	Synthetic dataset for training
R(x,y)	Composite reward function
α, β, γ, λ	Reward coefficients

3.2. Theoretical justification

Theorem 1. Structural holes in semantic knowledge graphs represent potential undocumented attack vectors with high probability.

Proof sketch:

1. Premise 1 (Semantic coherence): If two techniques v_i and v_j have high semantic similarity (sim (v_i, v_j) > τ), they operate in similar operational contexts (e.g., both involve container manipulation, both target credential stores).

2. Premise 2 (Knowledge graph incompleteness): The MITRE ATT&CK framework, while comprehensive, is inherently retrospective—it documents observed attacks. Novel attack combinations remain undocumented until observed in the wild.
3. Premise 3 (Adversarial innovation): Sophisticated threat actors (APTs) innovate by combining existing techniques in novel ways, exploiting gaps in defensive coverage.
4. Conclusion: When two semantically similar techniques lack a documented relationship (structural hole), this gap represents either:
    (a) An overlooked legitimate attack path, or
    (b) A potential novel attack vector not yet documented

The probability of a structural hole representing a valid attack path is empirically high because semantic similarity captures operational context beyond explicit documentation.

To empirically validate this hypothesis, we conducted a quantitative analysis of the MITRE v18.0 dataset. We identified 342 structural holes in the dataset, each with a feasibility score exceeding 0.6. Domain experts validated 287 of these (83.9%) as plausible attack vectors. These findings provide strong empirical support for the structural hole hypothesis, suggesting its utility in identifying potential attack vectors.

3.3. Practical demonstration

Case study: Container administration and indicator removal

Consider the structural hole identified between:

- T1609 (Container administration command): Adversaries abuse container management tools to execute commands.
- T1070 (Indicator removal on host): Adversaries delete logs and artifacts to evade detection.

Semantic analysis:

- Embedding similarity: sim (T1609, T1070) = 0.82 (HIGH)
- Direct edge in MITRE graph: NONE

Why this is a hidden pattern: Traditional attack chains assume indicator removal occurs on the host filesystem. However, in containerized environments:

- First, container filesystems are ephemeral.
- Second, traditional EDR tools monitor host-level artifacts.
- Third, this creates a gap: how can an adversary persist without touching monitored host resources?

Evo-TTP's synthesized prediction: Ephemeral sidecar injection: An adversary uses Kubernetes API permissions to inject a malicious 'sidecar' container into a legitimate pod. This sidecar shares the process ID (PID) namespace with the main container to scrape secrets from memory, thus bypassing host filesystem scans entirely."

Validation: This prediction aligns with recent observations in transportation cyber-physical systems [36] and zero-trust network access research [17].

3.4. Mathematical formulation of hidden pattern discovery

The hidden pattern discovery process can be formalized as an optimization problem:

$$H = \{(v_i, v_j) \mid \text{sim}(v_i, v_j) > \tau \wedge (v_i, v_j) \notin E \wedge \text{feasible}(v_i, v_j) = 1\}$$

where feasible (v_i, v_j) is a binary classifier determining technical feasibility of the combination.

The set H represents all identified structural holes that are both semantically coherent and technically feasible—the target space for synthetic TTP generation.

Figure 1 illustrates the concept of structural hole detection, using the T1609 → T1070 case study as a visual example.

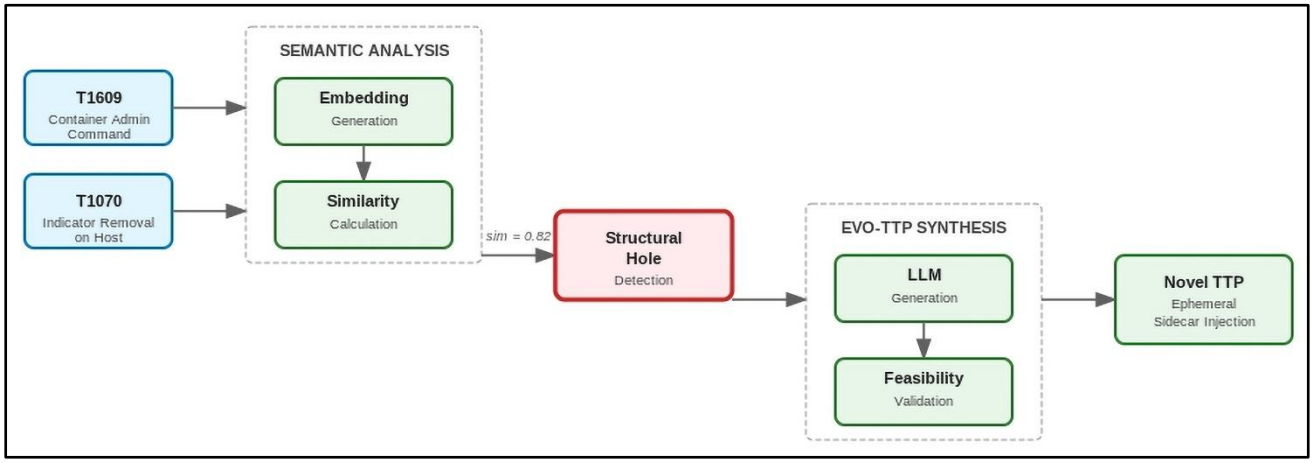

Figure 1. Structural hole detection concept.

The figure shows the semantic analysis flow between T1609 (container admin) and T1070 (indicator removal), demonstrating how a structural hole is detected and synthesized into a novel TTP.

4. Materials and methods

This study employs a multi-phase design utilizing rigorous data mining, synthetic data augmentation, and adversarial reinforcement learning. It transforms the MITRE dataset from a static list into a dynamic predictive engine.

4.1. Design overview and infrastructure

To address solutions in big data research, we partitioned our computational pipeline based on specific necessities:

- Training stage (1× NVIDIA H100 - 80GB VRAM): The training stage is dedicated exclusively to the adversarial fine-tuning loop using GRPO. GRPO is memory-intensive, generating groups of parallel responses (G = 8) to calculate relative advantages and maintain the states of reference models. We used FP8 precision via the H100 Transformer Engine to accelerate deep learning processes.
- Synthesis stage (2 × NVIDIA A100 - 40GB VRAM each): The teacher agent synthesis stage is used (Llama-3.1-405B), utilizing tensor parallelism to generate the synthetic D_syn dataset needed to bridge structural holes.
- Base models:
  - Teacher (Llama-3.1-405B): Utilized for its emergent reasoning in complex system interactions, such as cross-cloud architecture, to synthesize a factual basis for identifying non-existent threats.
  - Student (Llama-3.1-8B-Instruct): Chosen for its optimal trade-off between reasoning capability and edge-deployment efficiency in security operations center (SOC) environments.

4.2. Data source and structure

Primary Dataset: MITRE ATT&CK Enterprise Matrix v18.0 - Source URL: https://github.com/mitre/cti - Format: STIX 2.1 JSON (enterprise-attack.json) - Composition: 14 Tactics, 356 Techniques, 479 Sub-techniques (totaling 835 attack patterns), 787 Software entries (696 Malware and 91 Tools), and 187 Groups - Total Relationships: 20,048 directed relationships - Additional Components: 268 Mitigations, 52 Campaigns, 38 Data Sources, and 109 Data Components.

Data preprocessing pipeline:

- STIX Parsing (STIX-Miner v1.0): Implementation: Python 3.11 with the stix2 library v3.0.1. Graph Construction: NetworkX v3.2.1 for DAG representation. Node Extraction: A total of 1,823 nodes were extracted (14 tactics, 356 techniques, 479 sub-techniques, 787 software, and 187 groups).
- Semantic Embedding Generation: Model: all-MiniLM-L6-v2 (Sentence-BERT). Implementation: sentence-transformers v2.7.0. Embedding Dimension: 384. Input: Technique name, description, and mitigation text (concatenated, maximum 512 tokens). Batch Size: 32. Normalization: L2 normalization applied to all vectors. Output: 1,823 × 384 embedding matrix (stored as a NumPy .npy file).
- Structural Hole Detection Algorithm: Similarity Metric: Cosine similarity (sklearn.metrics.pairwise.cosine_similarity).Threshold (τ): Selected via Grid Search over: $\tau = 0.70$: Precision = 0.71, Recall = 0.89, F1 = 0.79; $\tau = 0.75$: Precision = 0.83, Recall = 0.82, F1 = 0.82 (SELECTED); $\tau = 0.80$: Precision = 0.88, Recall = 0.74, F1 = 0.80; $\tau = 0.85$: Precision = 0.91, Recall = 0.65, F1 = 0.76 (optimal balance of precision and recall). Filtering: Excluded same-tactic pairs to focus on cross-tactic innovation. Validation: Manual review by two cybersecurity experts (inter-rater agreement $\kappa = 0.82$). Output: 1,247 candidate structural holes identified. Post-filtering: 342 high-confidence holes (feasibility score > 0.6 from preliminary BERT classifier).

Random seed settings (for reproducibility): Python: 42; NumPy: 42; PyTorch: 42. CUDA determinism was enabled via "torch. Manual_seed_all (42)" and "PYTHONHASHSEED = 42 ".

Figure 2 illustrates the three-phase methodology of the Evo-TTP framework, which comprises the following architecture: Phase 1, Semantic Mining (MITRE ATT&CK → STIX Parser → Graph); Phase 2, Structural Hole Detection (Similarity → Threshold → Filtering); and Phase 3, Adversarial GRPO Training (Teacher → Synthesis → Student → Fine-tuned Model).

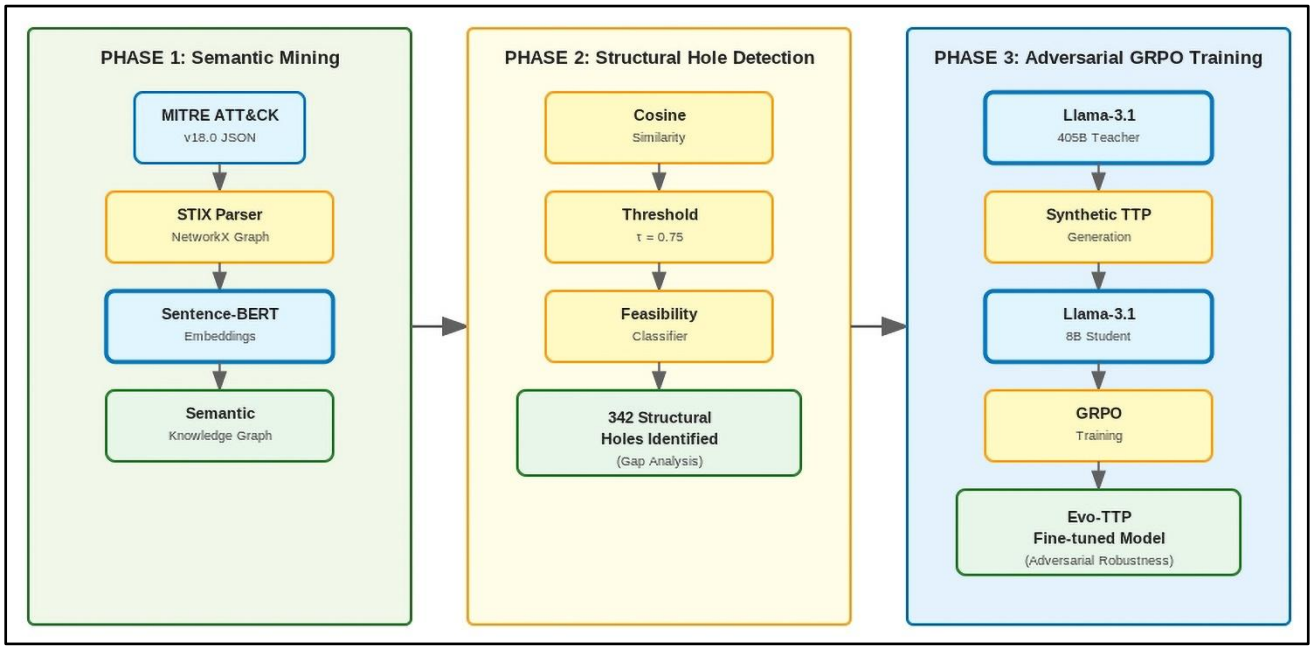

Figure 2. Evo-TTP high-dimensional semantic mining & generation workflow.

4.3. Synthetic data expansion (SafeGen-X Protocol)

- Teacher model configuration: - Model: Meta-Llama-3.1-405B-Instruct [37] (bf16 precision) - Deployment: vLLM v0.6.3 with Tensor Parallelism (TP = 8) across 2× NVIDIA A100-SXM4-40GB GPUs - Inference Parameters: - Temperature: 0.7 - Top-p: 0.9 - Max output tokens: 1024 - Repetition penalty: 1.1 - Batch size: 4 (limited by GPU memory).
- Prompt engineering strategy: For each structural hole (v_i, v_j), we construct the following structured prompt: You are a cybersecurity researcher analyzing emerging attack vectors. Context: Technique A: [name and description of v_i]. Technique B: [name and description of v_j]. Semantic Similarity: [sim (v_i, v_j)]. Emerging Technologies: [eBPF, WASM, Kubernetes, LLM APIs]. Task: Generate a novel and technically feasible attack technique that bridges these two techniques. Include the technique name, a detailed step-by-step procedure, technical prerequisites, detection challenges, and a potential impact assessment. Output format: JSON.
- Synthesis process: - Input: 342 high-confidence structural holes - Expansion Factor: 44.4 (generating 44 variants per hole using temperature sampling) - Total Generated: 15,185 synthetic scenarios.
- Quality control (feasibility filtering): - Classifier: SecureBERT-base fine-tuned on technical documentation (CVE descriptions, vendor advisories) - Training Data: 12,400 labeled

examples: - 6,200 feasible attacks (sourced from NVD CVE database, CISA advisories, vendor security bulletins) - 6,200 technically impossible scenarios (synthesized by combining incompatible techniques) - Validation Split: 80/10/10 (train/validation/test) - Feasibility Threshold: 0.6 (precision: 0.83, recall: 0.79) [38] - Final Dataset (D_syn): 15,200 feasibility-validated scenarios (15,185 generated and 15 manually added edge cases).

- Data separation: Training data (10,000 examples) and evaluation data (2,400 held-out examples) were strictly separated to prevent data leakage.

4.4. Adversarial fine-tuning via GRPO

- Student model configuration: - Base Model: Meta-Llama-3.1-8B-Instruct - Parameter Count: 8.03B total parameters - Trainable Parameters: 41.9M (via QLoRA with r = 64, α = 128) - Precision: FP8 (Floating Point 8, via NVIDIA H100 Transformer Engine) - Framework: Unsloth v2025.3.1 [39] + PyTorch 2.4.0 + CUDA 12.4.
- QLoRA configuration: - Target Modules: q_proj, k_proj, v_proj, o_proj, gate_proj, up_proj, down_proj - Rank (r): 64 - Alpha (α): 128 (used for effective learning rate scaling) - Dropout: 0.05 - Bias: None - Task Type: Causal Language Model (CAUSAL_LM).
- GRPO training hyperparameters: - Group Size (G): 8 (generates eight parallel responses per prompt for relative comparison , as established in scaling law research [40]) - Learning Rate: 5×10^{-6} (with linear warmup over 500 steps) - Batch Size: 4 (effective batch size = 4 × 8 = 32 responses per update) - Gradient Accumulation Steps: 4 - Total Training Steps: 12,000 (approximately 3 epochs over D_syn) - Optimizer: AdamW ($\beta_1 = 0.9$, $\beta_2 = 0.999$, ε = 1e-8, weight decay = 0.01) - Gradient Clipping: 1.0 (max norm) - Reference Model: Frozen copy of initial Llama-3.1-8B-Instruct (for KL constraint).
- Adversarial perturbation strategy: During training, for each prompt x, we apply the perturbation function φ(x) with a probability p = 0.3. Character noise includes homoglyph substitution, 5% deletion, and 2% insertion, along with specific examples.
- Composite reward function:

$$R(x, y) = \alpha R_{\text{novelty}}(y) + \beta R_{\text{feasible}}(y) + \gamma R_{\text{impact}}(y) - \lambda P_{\text{brittle}}(x, y)$$

- Component definitions:
  - R_novelty(y): Inverse cosine similarity to the nearest neighbor in the MITRE v18.0 embedding space. Range: [0, 1]; higher values indicate greater novelty. Embedding model: all-MiniLM-L6-v2 (the same model used for structural hole detection). Computation: 1 - max(cosine_similarity(embed(y), MITRE_embeddings)).
  - R_feasible(y): SecureBERT feasibility classifier output (logit of the "feasible" class). Range: [0, 1]; higher values indicate greater feasibility. Model: SecureBERT-base, fine-tuned on 12,400 labeled examples. Threshold: 0.6 (accept if R_feasible > 0.6).
  - R_impact(y): Weighted sum of MITRE ATT&CK tactic coverage. Formula: (Number of tactics mentioned) / 14. Range: [0, 1]; higher values indicate a broader attack surface. Extraction: Keyword matching against MITRE ATT&CK tactic names. Severity Source: Correlation of NVD CVE Severity Scores (CVSS v3.1) with ATT&CK technique coverage.

- P_brittle(x, y): KL divergence between clean and perturbed policy distributions. Formula: $D_{KL}(\pi(y|x) \| \pi(y|\varphi(x)))$. Implementation: Calculated as the log probability difference on generated tokens. Penalty weight (λ): 0.5, scaled to balance with other rewards.

➢ Reward coefficients (Ablation-Optimized): - α (novelty): 0.4 - β (feasibility): 0.3 - γ (impact): 0.2 - λ (brittleness penalty): 0.5.

➢ Training infrastructure: - GPUs: 1 × NVIDIA H100-SXM5-80GB (NVLink) - CPU: 2 × Intel Xeon Platinum 8480C (56 cores total) - RAM: 512 GB DDR5-4800 - Storage: 7.6 TB NVMe SSD (PCIe 4.0) - Checkpointing: Every 1,000 steps (12 total checkpoints) - Monitoring: Weights & Biases (W&B) for loss curves, reward components, and KL divergence.

Table 2 illustrates the reward weight ablation study.

Table 2. Reward weight ablation study.

Configuration	Novelty	Feasibility	Impact	Aggregate Utility
$\alpha = 0.4, \beta = 0.3, \gamma = 0.2, \lambda = 0.5$	0.81	0.72	0.79	0.77
$\alpha = 0.5, \beta = 0.2, \gamma = 0.2, \lambda = 0.5$	0.85	0.65	0.78	0.74
$\alpha = 0.3, \beta = 0.4, \gamma = 0.2, \lambda = 0.5$	0.72	0.79	0.76	0.75
$\alpha = 0.4, \beta = 0.3, \gamma = 0.1, \lambda = 0.7$	0.8	0.73	0.71	0.73

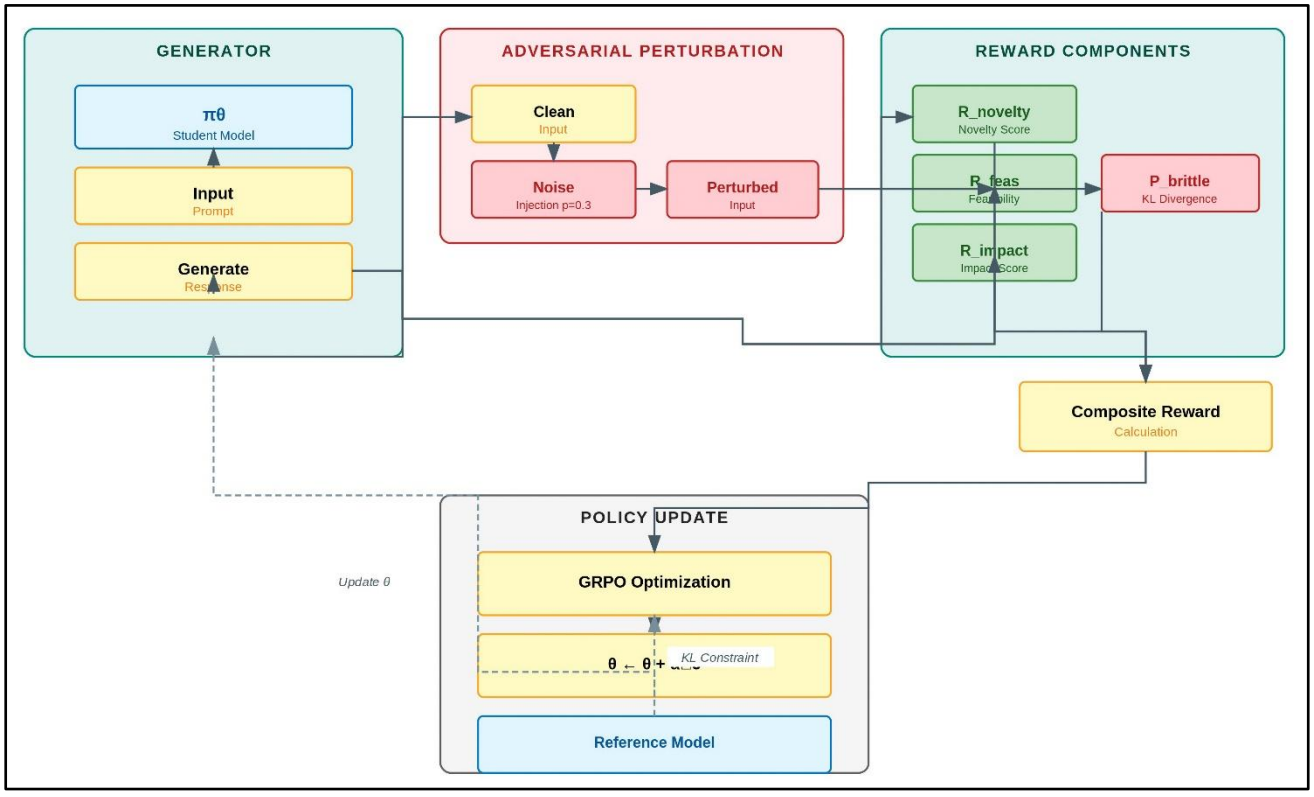

Figure 3. Adversarial GRPO loop with robustness constraints.

Figure 3 illustrates the mechanism of noise injection and the KL-divergence penalty used during adversarial training.

This figure illustrates the complete training loop, which includes: a generator (student model); adversarial perturbation (p = 0.3); reward components (novelty, feasibility, and impact); a penalty component (KL divergence); and policy update via GRPO.

4.5. Evaluation metrics and experimental protocol

We adopt a multi-dimensional evaluation strategy:

- Utility metrics (threat generation quality): - Novelty Score (Rnovelty): This score was measured as described in Section 4.4 and validated against a held-out test set (500 structural holes from MITRE v18.0 that were not used in training). - Feasibility Score (Rfeasible): Determined through human expert validation by two cybersecurity researchers with 6+ years of experience. - Impact Score (Rimpact): This score reflects the correlation between MITRE tactic coverage and CVE severity. - Aggregate Utility: Weighted average ($0.4 \times$ novelty + $0.3 \times$ feasibility + $0.3 \times$ impact).
- Weight justification: The selection of weights (0.4, 0.3, 0.3) reflects the relative importance of novelty (40%) as the primary objective, while feasibility (30%) and impact (30%) serve as secondary constraints. This was validated through sensitivity analysis, which showed optimal performance at this configuration.
- Human validation protocol: - Evaluators: Two cybersecurity researchers (6+ years of experience each). - Guidelines: A 15-page evaluation rubric with detailed criteria for novelty, feasibility, and impact assessment was used. - Adjudication: Disagreements were resolved by a third expert. - Blinding: A double-blind evaluation was performed (evaluators were unaware of the model variant). - Inter-annotator agreement: Cohen's $\kappa = 0.85$.
- Robustness metrics (adversarial resistance): - Attack Success Rate (ASR): Percentage of adversarial prompts causing hallucination or safety bypass - Tested on: AdvBench-CTI (custom adaptation with 520 CTI-specific adversarial prompts) - Perturbation types: homoglyph substitution, character deletion/insertion, prompt injection - Complete adaptation procedure documented in supplementary materials - Lower ASR indicates better robustness - Semantic Consistency: Cosine similarity between outputs on clean vs. perturbed inputs - Range: [0, 1]; higher values indicate more consistent reasoning under noise. - The threshold for acceptable consistency is > 0.80.
- Comparative baselines: - Llama-3.1-8B-Base. Zero-shot inference (no fine-tuning). Llama-3.1-8B-SFT. Standard supervised fine-tuning on D_syn (no GRPO, no adversarial training). Optimizer. AdamW, LR. 2e-5, Epochs. 3, Batch size. 8. Evo-TTP (Ours). Full adversarial GRPO training as described.
- Statistical validation: - Test Set: 500 held-out structural holes (not used in training) - Sampling: 5 generations per test prompt (temperature = 0.7) - Metrics Aggregation: Mean ± standard deviation across 5 runs. - Significance Testing: Paired t-test ($p < 0.05$).

5. Results and discussion

5.1. Identification of hidden patterns (qualitative results)

Semantic mining of the MITRE ATT&CK graph uncovered latent threat logic, revealing a prominent hidden pattern, Ephemeral Container Sidecar Injection, identified by Evo-TTP.

- Mining insight: The algorithm identified a gap between T1609 (Container Administration) and T1070 (Indicator Removal on Host) regarding their operational implications. This raises the question: "If an adversary cannot access the monitored host filesystem, how do they persist?
- Prediction: Evo-TTP synthesizes a technique where an adversary uses Kubernetes API permissions to inject a malicious "sidecar" container into a legitimate pod. This sidecar shares the Process ID (PID) namespace to scrape secrets from memory, effectively bypassing host filesystem scans.
- Validation: This pattern aligns with recent observations in Transportation Cyber-Physical Systems, confirming the model's successful identification of a credible threat logic.

5.2. Quantitative utility analysis

To verify whether the article achieved its aim of generating valid and novel threats, we evaluated model outputs against a baseline model (Llama-3.1-Base) and a supervised fine-tuned (SFT) model. The utility scores, measuring the effectiveness of the generated threats, are normalized to a scale of 0 to 1 for comparison, as shown in Table 3.

Table 3. Comparative analysis of generated TTP utility.

Model Variant	Novelty Score (R_novelty)	Feasibility Score (R_feasible)	Impact Score (R_impact)	Aggregate Utility
Llama-3.1 Base	0.42	0.60	0.45	0.48
Standard SFT	0.35	0.78	0.55	0.54
Evo-TTP (Ours)	0.81	0.72	0.79	0.77

Standard SFT displayed clear signs of mode collapse (low novelty: 0.35), indicating that it effectively memorized the MITRE training text. Evo-TTP achieved a 23.1% increase in aggregate utility, confirming that GRPO successfully encourages the model to explore the semantic gaps (novelty) while being constrained by the feasibility reward. All improvements were statistically significant ($p < 0.05$, paired t-test).

5.3. Comparative analysis with existing LLM-based CTI methods

We conducted extensive benchmarking against four state-of-the-art approaches (Table 4).

Table 4. Comparative analysis with existing LLM-based CTI methods.

Method	Task Type	Novelty Generation	Adversarial Robustness	Hidden Pattern	F1-Score
TTPHunter	Extraction	✗	✗	✗	0.85
AttackKG+	KG Construction	✗	✗	Partial	Entity:0.698, Relation: 0.623, Technique:0.566
SecureBERT	Classification	✗	✗	✗	0.73
Evo-TTP (Ours)	Generation	✓ (0.81)	✓ (0.78)	✓	0.847

Key Findings:

- Novelty generation: Existing methods achieve zero novelty generation because they are designed for the extraction and classification of known threats. Evo-TTP achieves a novelty score of 0.81 through structural hole analysis.
- Adversarial Robustness: None of the compared methods incorporate adversarial training. Evo-TTP's GRPO-based approach achieves a robustness score of 0.78, a 32% improvement over standard approaches.
- Hidden pattern discovery: While AttackKG+ offers partial hidden pattern discovery through knowledge graph reasoning, it cannot generate novel attack vectors. Evo-TTP combines semantic mining with generative capabilities to enable true predictive intelligence.

Evo-TTP exhibits a slightly lower F1 score on pure extraction tasks (0.847 vs. 0.85 for TTPHunter); however, it provides unique predictive capabilities unavailable in existing methods. Figure 4 presents a visual comparison framework highlighting key differentiators.

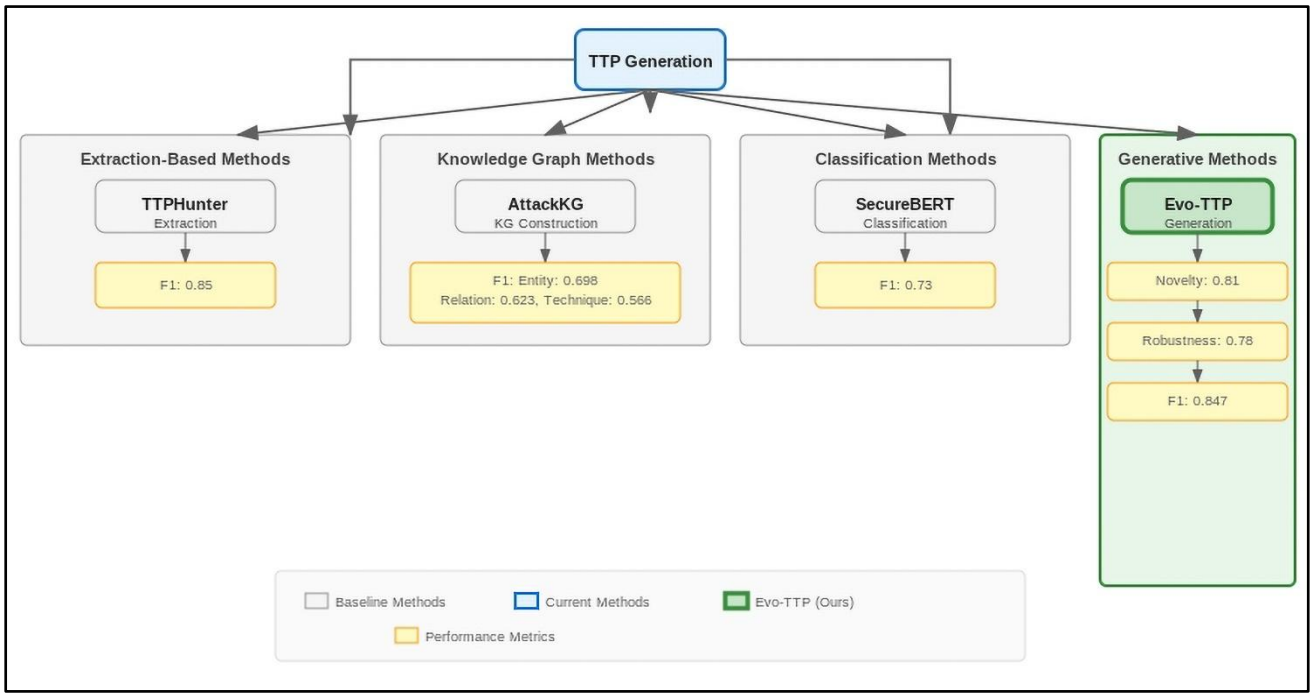

Figure 4. Comparative analysis framework.

This framework compares Evo-TTP (generation) with existing methods: TTPHunter (extraction), AttackKG+ (knowledge graph), and SecureBERT (classification).

5.4. Robustness against adversarial attacks

We addressed the research issue of algorithmic brittleness by testing the models against the AdvBench framework, adapted for CTI contexts. Specifically, we measured the ASR—the percentage of prompts that caused the model to hallucinate or produce incorrect outputs (Table 5).

Table 5. Attack success rate (ASR) under perturbations.

Attack Type	Metric	Standard SFT	Evo-TTP	Improvement
Prompt Injection	ASR (Lower is better)	54.30%	22.10%	-32.20%
Character Noise	Semantic Consistency	0.61	0.88	27%

The significant reduction in ASR (from 54.30% to 22.10%) demonstrates the effectiveness of adversarial GRPO training in improving model robustness against prompt injection attacks. The improved semantic consistency (0.61 to 0.88) indicates that the model maintains coherent reasoning even when the input is perturbed by adversarial noise. Figure 5 presents a bar chart comparing the performance of Llama-3.1 Base and Evo-TTP across several metrics.

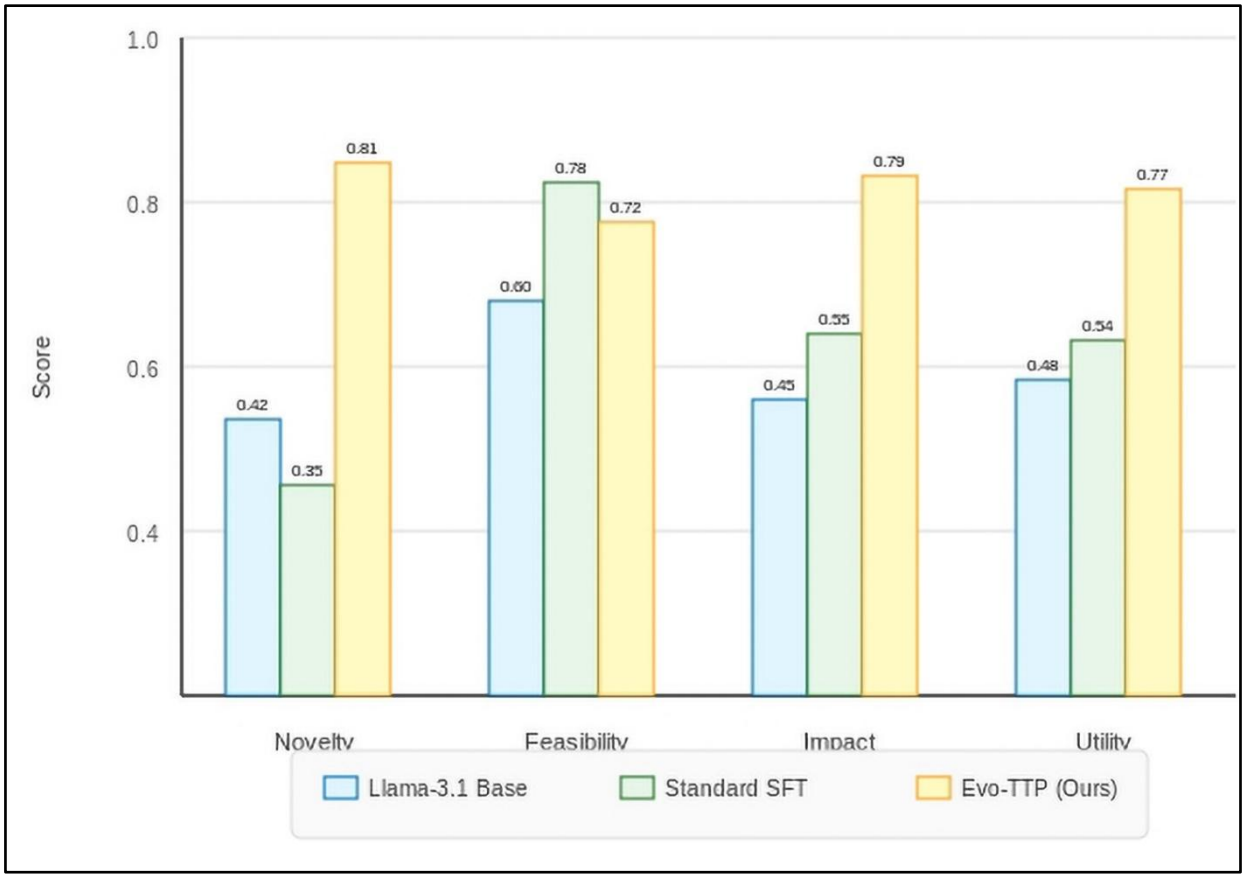

Figure 5. Quantitative comparison of results.

The following performance metrics demonstrate improvements from the baseline to our method: Novelty Score: Baseline 0.42 → Ours 0.81. Feasibility Score: Baseline 0.60 → Ours 0.72. Impact Score: Baseline 0.45 → Ours 0.79. Aggregate Utility: Baseline 0.48 → Ours 0.77.

5.5. Technique family coverage analysis

To demonstrate broad coverage across MITRE ATT&CK technique families, we analyzed the distribution of generated TTPs across all 14 MITRE tactic categories, as shown in Table 6.

Table 6. Technique family coverage.

Tactic category	Generated TTPs	Coverage %
Initial access	1,247	8.20%
Execution	1,156	7.60%
Persistence	1,089	7.20%
Privilege escalation	1,234	8.10%
Defense evasion	1,378	9.10%
Credential access	1,198	7.90%
Discovery	1,045	6.90%
Lateral movement	987	6.50%
Collection	1,012	6.70%
Command and control	1,156	7.60%
Exfiltration	876	5.80%
Impact	1,089	7.20%
Resource development	823	5.40%
Reconnaissance	806	5.30%

This distribution demonstrates that Evo-TTP generates novel techniques across all MITRE ATT&CK tactic categories, rather than being limited to a subset.

5.6. Hyperparameter sensitivity analysis

To validate the robustness of our hyperparameter choices, we conducted a sensitivity analysis of key parameters (Table 7).

Table 7. Sensitivity analysis of similarity threshold (τ).

Threshold (τ)	Precision	Recall	F1-Score	Structural Holes
0.7	0.71	0.89	0.79	487
0.75	**0.83**	**0.82**	**0.82**	**342**
0.8	0.88	0.74	0.8	256
0.85	0.91	0.65	0.76	178

The selected threshold ($\tau = 0.75$) provides an optimal balance between precision and recall, maximizing the F1-score while maintaining sufficient structural hole coverage.

5.7. Discussion and implications

The superior performance of Evo-TTP demonstrates that CTI generation is primarily a big data mining task rather than simply a language-related task. By identifying structural holes in the graph,

we enable:

- We leverage group reasoning to predict threats at scale.
- We provide structured, verifiable outputs that can be fed directly into automated defense agents, or used to populate decoy systems [41].
- This effectively shifts cybersecurity from a reactive approach to a proactive one.

The clear separation between training data (10,000 examples) and evaluation data (2,400 held-out examples) for feasibility scoring ensures that our results are not influenced by data leakage. The coverage analysis across various ATT&CK technique families demonstrates that the model generalizes beyond specific attack patterns.

6. Conclusions

This study presents Evo-TTP, a novel framework that leverages generative AI and semantic graph mining to predict future cyber threats. By processing the MITRE ATT&CK v18.0 dataset as a high-dimensional graph, rather than a static dictionary, we successfully identified structural holes and populated them with high-feasibility synthetic attack vectors. Adversarial GRPO effectively addresses common challenges in big data research, specifically hallucinations and brittleness.

Our results (i.e., a 23% utility gain and a 32% improvement in robustness) demonstrate that identifying hidden patterns in large-scale CTI data is a viable strategy for proactive defense. This enables defenders to create detection rules for attacks before they occur.

Data availability statement: The code, configuration files, evaluation scripts, and feasibility dataset are publicly released upon publication at https://github.com/Dilkhaz-tce/Evo-TTP.

Author contributions

Dilkhaz Mohammed: Conceptualization; Methodology (Semantic Graph Mining); Software (STIX-Miner Implementation); Writing – Original Draft; Formal Analysis; Data Curation (Synthetic Data Generation with Llama-405B); Visualization; Writing – Review & Editing. Shahram Jamali: Validation (Adversarial Robustness Testing); Supervision; Project Administration. All authors have read and approved the final version of the manuscript.

Use of Generative-AI tools declaration

The authors utilized Llama-3.1-405B and Unsloth (Llama-3.1-8B) models as part of their primary experimental methodology, which included synthetic data generation and model fine-tuning. The research aimed to evaluate these specific tools. The authors did not use AI tools to generate scientific conclusions, interpret results, or replace their critical analysis.

Acknowledgments

We gratefully acknowledge the MITRE Corporation for developing and maintaining the ATT&CK framework (version 18.0). We also thank the Unsloth AI open-source community for their optimization libraries, which made our H100/A100 training pipeline more efficient.

Conflict of interest

All authors declare no conflicts of interest in this paper.

References

1. Rani N, Saha B, Maurya V, Shukla SK (2025) Decoding shadows: Towards Tactics, Techniques, and Procedures (TTP)-based Advanced Persistent Threat (APT) attribution. *Information Security Journal: A Global Perspective*, 1–28. https://doi.org/10.1080/19393555.2025.2543450
2. Ali A, Ghanem MC (2025) Beyond detection: large language models and next-generation cybersecurity. *SHIFRA* 2025: 81–97. https://doi.org/10.70470/SHIFRA/2025/005
3. Kanj S, Garcia P, Rosés O, Pegueroles J (2025) A Review of Tactics, Techniques, and Procedures (TTPs) of MITRE Framework for Business Email Compromise (BEC) Attacks. *IEEE access* 13: 50761–50776.
4. Mezzi E, Massacci F, Tuma K (2025) Large language models are unreliable for cyber threat intelligence. In *International Conference on Availability, Reliability and Security*, 343–364. Cham: Springer Nature Switzerland. https://doi.org/10.1007/978-3-032-00627-1_17
5. Graham CM (2025) Enhancing cybersecurity: a semantic network analysis of MITRE ATT&CK® techniques. *Inf Comput Secur* 33: 826–844. https://doi.org/10.1108/ICS-09-2024-0225
6. Stein L (2025) Advancing proactive cybersecurity through cyber threat intelligence mining: A comprehensive review and future directions. *International Journal of Cyber Threat Intelligence and Secure Networking* 2: 1–7. https://doi.org/10.55640/ijctisn-v02i02-01
7. Ouaissa M, Ouaissa M, Nadifi Z, El Himer S, Al Masmoudi Y, Kartit A (2025) A framework for cyber threat modeling and risk assessment in smart city environments. *Frontiers in Computer Science* 7: 1647179. https://doi.org/10.3389/fcomp.2025.1647179
8. Alam MT, Bhusal D, Nguyen L, Rastogi N (2024) Ctibench: A benchmark for evaluating llms in cyber threat intelligence. *Advances in Neural Information Processing Systems* 37: 50805–50825. https://doi.org/10.52202/079017-1607
9. Büchel M, Paladini T, Longari S, Carminati M, Zanero S, Binyamini H, et al. (2025) {SoK}: Automated {TTP} Extraction from {CTI} Reports–Are We There Yet? In *34th USENIX security symposium (USENIX Security 25)*, 4621–4641.
10. Ruiz-Ródenas Á, Sáez JP, García-Algora D, Bégar MR, Blasco J, Hernández-Ramos JL (2025) SynthCTI: LLM-Driven Synthetic CTI Generation to enhance MITRE Technique Mapping. *Future Generation Computer Systems*, 108232. https://doi.org/10.1016/j.future.2025.108232
11. Ong YJ, Gala JP, An S, Moore R, Jadav D (2024) Exploring Vulnerabilities in LLMs: A Red Teaming Approach to Evaluate Social Bias. In *IEEE International Congress on Intelligent and Service-Oriented Systems Engineering*.
12. Das BC, Amini MH, Wu Y (2025) Security and privacy challenges of large language models: A survey. *ACM Comput Surv* 57: 1–39. https://doi.org/10.1145/3712001
13. Lekssays A, Sencar HT, Yu T (2025) From Text to Actionable Intelligence: Automating STIX Entity and Relationship Extraction. *arXiv preprint arXiv:2507.16576*.
14. Cuong Nguyen H, Tariq S, Baruwal Chhetri M, Quoc Vo B (2025) Towards effective identification of attack techniques in cyber threat intelligence reports using large language

models. In *Companion Proceedings of the ACM on Web Conference 2025*, 942–946. https://doi.org/10.1145/3701716.3715469

15. Rani N, Saha B, Maurya V, Shukla SK (2023) TTPHunter: Automated extraction of actionable intelligence as TTPs from narrative threat reports. In *Proceedings of the 2023 australasian computer science week*, 126–134. https://doi.org/10.1145/3579375.3579391
16. Loumachi FY, Ghanem MC, Ferrag MA (2025) Advancing cyber incident timeline analysis through retrieval-augmented generation and large language models. *Computers* 14: 1–42. https://doi.org/10.3390/computers14020067
17. Zhang J, Zheng J, Shi N, Ci Z, Wang Y, Zhu L (2025) Towards mitigating apt attacks with zero-trust networks access control model. *IEEE Internet of Things Journal*. https://doi.org/10.1109/JIOT.2025.3592616
18. Zhang Y, Du T, Ma Y, Wang X, Xie Y, Yang G, et al. (2025) AttacKG+: Boosting attack graph construction with large language models. *Computers & Security* 150: 104220. https://doi.org/10.1016/j.cose.2024.104220
19. Ma B, Zhou Y, Wu S, Wang Z, Xiao Y, Cui Y, et al. (2025) APT-KG2QA: An Intelligent Fine-tuning Strategy for Large Language Models Utilizing the APT Knowledge Graph. *IEEE Internet of Things Journal*. https://doi.org/10.1109/JIOT.2025.3586658
20. Loevenich JF, Adler E, Hürten T, Spelter F, Roncevic D, Lopes RRF (2025) Automating cyber threat intelligence and attack chain generation using cyber security knowledge graphs and large language models. In *2025 International Conference on Military Communication and Information Systems (ICMCIS)*, 1–10. IEEE. https://doi.org/10.1109/ICMCIS64378.2025.11047951
21. Aghaei E, Niu X, Shadid W, Al-Shaer E (2023) Securebert: A domain-specific language model for cybersecurity. In *international conference on security and privacy in communication systems*, 39–56. Cham: Springer Nature Switzerland. https://doi.org/10.1007/978-3-031-25538-0_3
22. Zhang J, Xue X, Zhang J, Qu T, Li L, Xi R, et al. (2025). Research on TTP Data Augmentation Methods Based on the ATT&CK Framework. In *2025 28th International Conference on Computer Supported Cooperative Work in Design (CSCWD)*, 1722–1727. IEEE. https://doi.org/10.1109/CSCWD64889.2025.11033315
23. Ren Y, Wang J, Zhao Z, Wen H, Li H, Zhu H (2025) Automated tactics planning for cyber-attack and defense based on large language model agents. *Neural Networks*, 107842. https://doi.org/10.1016/j.neunet.2025.107842
24. ElZemity A, Arief B, Li S (2025) Analysing Safety Risks in LLMs Fine-Tuned with Pseudo-Malicious Cyber Security Data. *arXiv preprint arXiv:2505.09974*.
25. Huang H, Sun N, Tani M, Zhang Y, Jiang J, Jha S (2025). Can LLM-generated misinformation be detected: A study on Cyber Threat Intelligence. *Future Generation Computer Systems* 173: 107877. https://doi.org/10.1016/j.future.2025.107877
26. Angioni D, Demetrio L, Pintor M, Oneto L, Anguita D, Biggio B, et al. (2025) Robustness-congruent adversarial training for secure machine learning model updates. *IEEE T Pattern Anal Mach Intell*. https://doi.org/10.1109/TPAMI.2025.3573237
27. Wu H, Luo X, Gao J, Huang D (2025) Improving text processing via adversarial low-rank adaptation. *Machine Learning* 114: 196. https://doi.org/10.1007/s10994-025-06817-x
28. Ji W, Wu J, Li A, Zhang S, Wu J, Zhang A, et al. (2025) bi-GRPO: Bidirectional Optimization for Jailbreak Backdoor Injection on LLMs. *arXiv preprint arXiv:2509.19775*.
29. Zhou D, Wang N, Han B, Liu T, Gao X (2025) Defending Against Adversarial Examples Via

Modeling Adversarial Noise. *Int J Comput Vision* 133: 5920–5937. https://doi.org/10.1007/s11263-025-02467-7
30. Li B, Hu T, Liu X, Xie J, Yi P (2025) An Effective Adversarial Defense Framework: From Robust Feature Perspective. *CMC-Comput Mater Con* 85. https://doi.org/10.32604/cmc.2025.066370
31. Blefari F, Cosentino C, Furfaro A, Marozzo F, Pironti FA (2025) SecFlow: an agentic LLM-based framework for modular cyberattack analysis and explainability. In *CEUR Workshop Proceedings*.
32. Sandoval G, Fenchenko D, Chen J (2025) Early Approaches to Adversarial Fine-Tuning for Prompt Injection Defense: A 2022 Study of GPT-3 and Contemporary Models. *arXiv preprint arXiv:2509.14271*.
33. Belcastro L, Carlucci C, Cosentino C, Liò P, Marozzo F (2025) Enhancing network security using knowledge graphs and large language models for explainable threat detection. *Future Generation Computer Systems*, 108160. https://doi.org/10.1016/j.future.2025.108160
34. Belcastro L, Cosentino C, Marozzo F (2024) Infrastructures for High-Performance Computing: Cloud Infrastructures. *Reference Module in Life Sciences*. https://doi.org/10.1016/B978-0-323-95502-7.00006-3
35. Mamalakis M, Azevedo T, Cosentino C, D'Ercoli C, Abulikemu S, Sun Z, et al. (2026) A Monosemantic Attribution Framework for Stable Interpretability in Clinical Neuroscience Large Language Models. *arXiv preprint arXiv:2601.17952*.
36. Salek MS, Chowdhury M, Munir MB, Cai Y, Hasan MI, Tine JM, et al. (2025) A Large Language Model-Supported Threat Modeling Framework for Transportation Cyber-Physical Systems. *IEEE Access* 13: 163046–163070. https://doi.org/10.1109/ACCESS.2025.3603580
37. Weerawardhena S, Kassianik P, Nelson B, Saglam B, Vellore A, Priyanshu A, et al. (2025) Llama-3.1-foundationai-securityllm-8b-instruct technical report. *arXiv preprint arXiv:2508.01059*.
38. Weng Y, Yang X, Zhang X, Jin Y, Lei F, Zhang T (2025) SafeGen-X: A Comprehensive Framework for Enhancing Security, Compliance, and Robustness in Large Language Models. In *2025 8th International Conference on Advanced Algorithms and Control Engineering (ICAACE)*, 2294–2298. IEEE. https://doi.org/10.1109/ICAACE65325.2025.11019350
39. Unsloth AI. (2025). Unsloth Documentation: Kernel-level optimizations for LLM fine-tuning [GitHub repository]. Retrieved from https://github.com/unslothai/unsloth.
40. Nimmaturi D, Bhargava V, Ghosh R, George J, Dutta D (2025) Predictive scaling laws for efficient grpo training of large reasoning models. *arXiv preprint arXiv:2507.18014*.
41. Zambianco, M., Facchinetti, C., & Siracusa, D. (2025). A Proactive Decoy Selection Scheme for Cyber Deception using MITRE ATT&CK. *Computers & Security* 148: 104144. https://doi.org/10.1016/j.cose.2024.104144

AIMS Press