MetaCRiSP 2026 | IEEE S&P | May 21, 2026

Evaluating LLMs Reliability for Coding

Adversarial Machine Learning Research Papers

Madhav Khanal · Jasser Jasser · Mina Basirat

Rollins College (Khanal, Jasser)

University of Central Florida (Basirat)

Why This Paper?

The problem

Meta-research in security requires systematically coding large paper corpora: methodology, evaluation practices, threat models. Manual coding is slow and expensive; most studies stay at tens of papers.

How this paper started

This question arose while we were manually coding adversarial ML papers for a separate meta-analysis. The process was slow enough that we stopped and asked: can LLMs do this reliably instead?

Our question

Can LLMs reliably automate the coding of security research papers? Specifically for adversarial ML, where threat models, evaluation methodology, and jargon are domain-specific. And: which coding variables work well under automation, and which do not?

Study Design

Papers

71 adversarial ML papers, 2013-2024. Selected based on citation by widely-used security artifacts: CleverHans, IBM ART, Foolbox, TextAttack, PyRIT, MITRE ATLAS.

Models

4 LLMs: GPT-4o, GPT-5.2, Gemini-3, Claude Sonnet-4.5. Each model ran 3 independent passes at temperature 0. One domain expert coded all papers once.

Variables

9 categorical variables in three groups: research characteristics (G1-G6), threat model (T1-T2), practical evaluation (Q1).

Metrics

Fleiss kappa for intra-rater consistency per model across 3 runs. Cohen kappa for every inter-rater pair. 95% confidence intervals via bootstrap.

What We Coded: 9 Categorical Variables

Research Characteristics (G1-G6)

G1	**Paper type**	Attack, Defense, Evaluation, Both, Attack/Defense, ...
G2	**Attack category**	Evasion, Poisoning, Privacy, Defense, N/A
G3	**Application domain**	Vision, NLP, LLMs, Audio, Malware, Tabular, Cross-domain
G4	**Publication venue**	ML, Security, Journal, arXiv-only
G5	**Code available**	Yes / No
G6	**Code release timing**	At-pub, Post-pub, Never

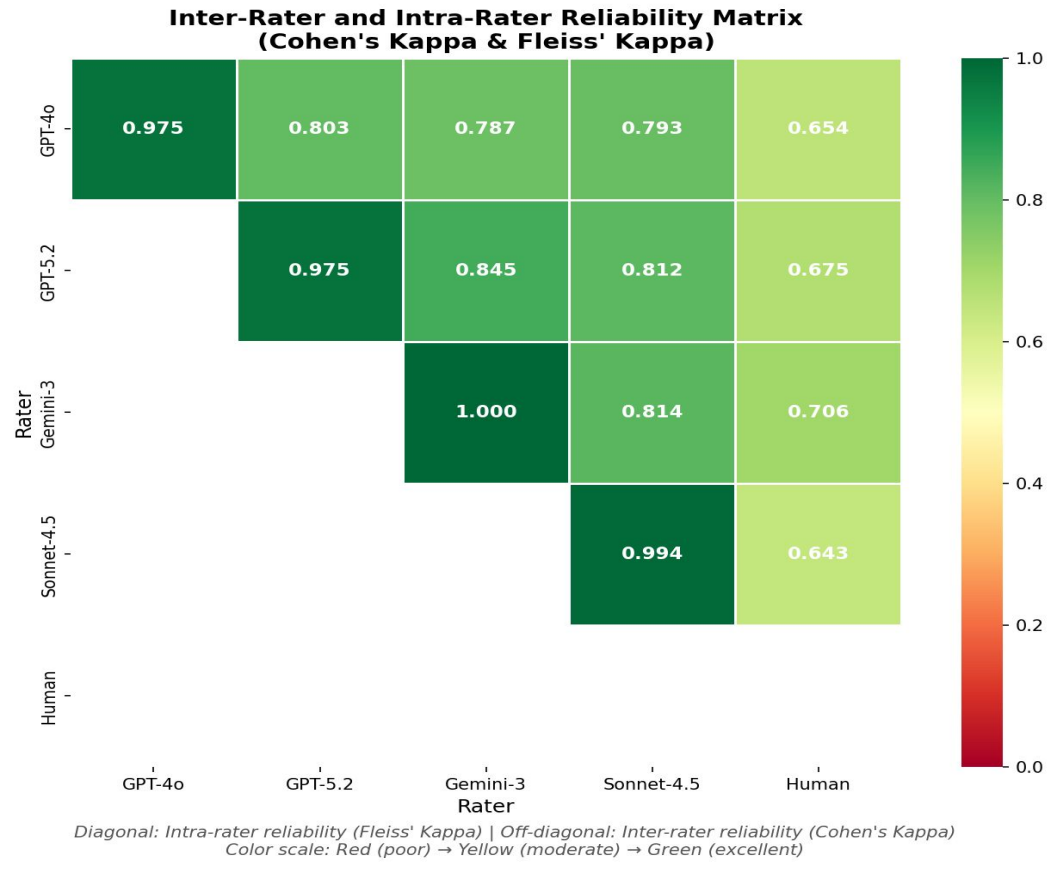
Threat Model (T1-T2, attack papers only)

T1	**Adversary model access**	White, Black, Gray, White/Black, N/A
T2	**Gradient required**	Yes, No, N/A

Practical Evaluation (Q1)

Q1	**Evaluation on a real-world system**	Yes, No, Partial, N/A

Overall Agreement



0.97 – 1.00

Intra-rater (all 4 LLMs)

Nearly identical outputs across 3 runs

0.79 – 0.85

Inter-LLM agreement

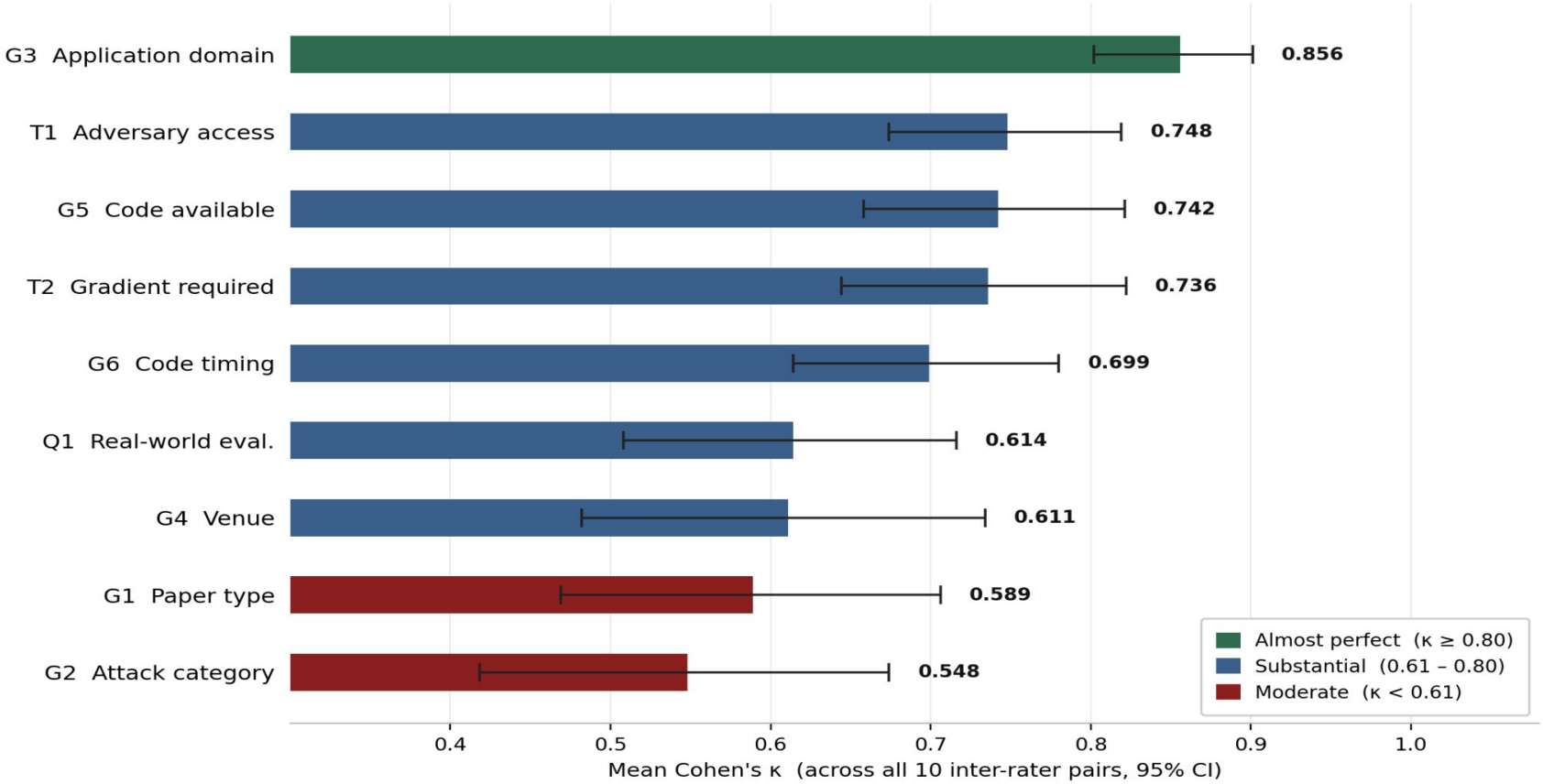
High convergence between model pairs

0.64 – 0.71

LLM vs. human coder

Substantial range per Landis & Koch

Reliability Variables



What Explains the Gap?

High agreement: pattern matching

G3 Domain Concrete, repeated lexical cues: "ImageNet", "CIFAR-10", "CNN", "BERT". Domain maps directly to surface vocabulary.

G5 Code available Stable forms: "GitHub", "code available at". Binary and unambiguous.

T2 Gradient required Explicit markers: gradient symbol, "backpropagation", "white-box attack".

Moderate agreement: interpretive judgment

G1 Paper type AutoAttack is both an attack method and an evaluation benchmark. Labeling it correctly requires understanding the author's primary framing, not just surface content.

G2 Attack category A poisoning attack that extracts training data could also be a privacy attack. The codebook allows overlapping categories, which shifts the ambiguity rather than resolving it.

Limitations

Single coder

Only one expert coded the 71 papers at submission. **Post-acceptance update:** a two-coder study on all 71 papers with independent third-rater validation found primary human-human mean κ = 0.818. Variable difficulty rankings were consistent: G1 remained the hardest for humans too (κ = 0.575), and G3 domain reached κ = 0.895 for human coders, comparable to LLMs at 0.856. Notably, the third independent rater agreed with primary coders at κ = 0.527–0.559, below LLM-human agreement of 0.64–0.71.

Model coverage

We tested four models due to API budget constraints. Reasoning-focused models such as o3 were not evaluated. Results reflect the state of these specific models in early 2026 and may shift as models evolve.

Prompt design

Each model received a single one-shot prompt with no per-model tuning. Few-shot examples or chain-of-thought prompting could improve reliability, particularly on interpretive variables.

Generalizability

Our corpus covers adversarial ML, selected by artifact adoption. Results may not generalize to other security subfields such as systems security or cryptography, where domain vocabulary and evaluation norms differ.

Takeaways

1. Some Reliability (Structured)

LLMs are a useful first pass for structured coding tasks.

Variables with stable lexical signals (Domain, Venue, Access Type) can be pre-coded at scale with substantial reliability.

2. Human Necessity (Interpretive)

Human review is still required for interpretive variables.

Conceptual ambiguity leads to inconsistent LLM performance. LLM output is a starting point, not ground truth.

3. Can LLMs Accelerate Research Audits?

Could structural patterns and overclaims in adversarial ML research such as reliance on "obfuscated gradients" or lack of testing on real deployed systems be potentially be accelerated with LLM assistance?

- Carlini & Wagner: Defensive distillation offered almost no real protection.
- Athalye et al.: 7 of 9 defenses at ICLR 2018 relied on obfuscated gradients.
- Apruzzese et al.: 80% of top security papers don't test on real systems.

Codebook, prompts, analysis scripts, and raw model outputs will be made publicly available upon acceptance.