Chasing Shadows: Pitfalls in LLM Security Research

Jonathan Evertz*[†], Niklas Risse*[‡], Nicolai Neuer[§], Andreas Müller[¶], Philipp Normann[‖], Gaetano Sapia[‡], Srishti Gupta[#], David Pape[†], Soumya Shaw[†], Devansh Srivastav[†], Christian Wressnegger[§], Erwin Quiring[◇], Thorsten Eisenhofer[†], Daniel Arp[‖], Lea Schönherr[†]

[†] CISPA Helmholtz Center for Information Security [‡] Max Planck Institute for Security and Privacy

[§] Karlsruhe Institute of Technology [¶] Ruhr University Bochum [‖] TU Wien

[#] Sapienza University of Rome [◇] _fbeta

* Equal contribution

arXiv

1,156 +42% YoY

cs.CR — Cryptography & Security · submissions · Oct 2026



AI Impact Vertical lines: major AI model releases

vester.si/ai-impact/arxiv/

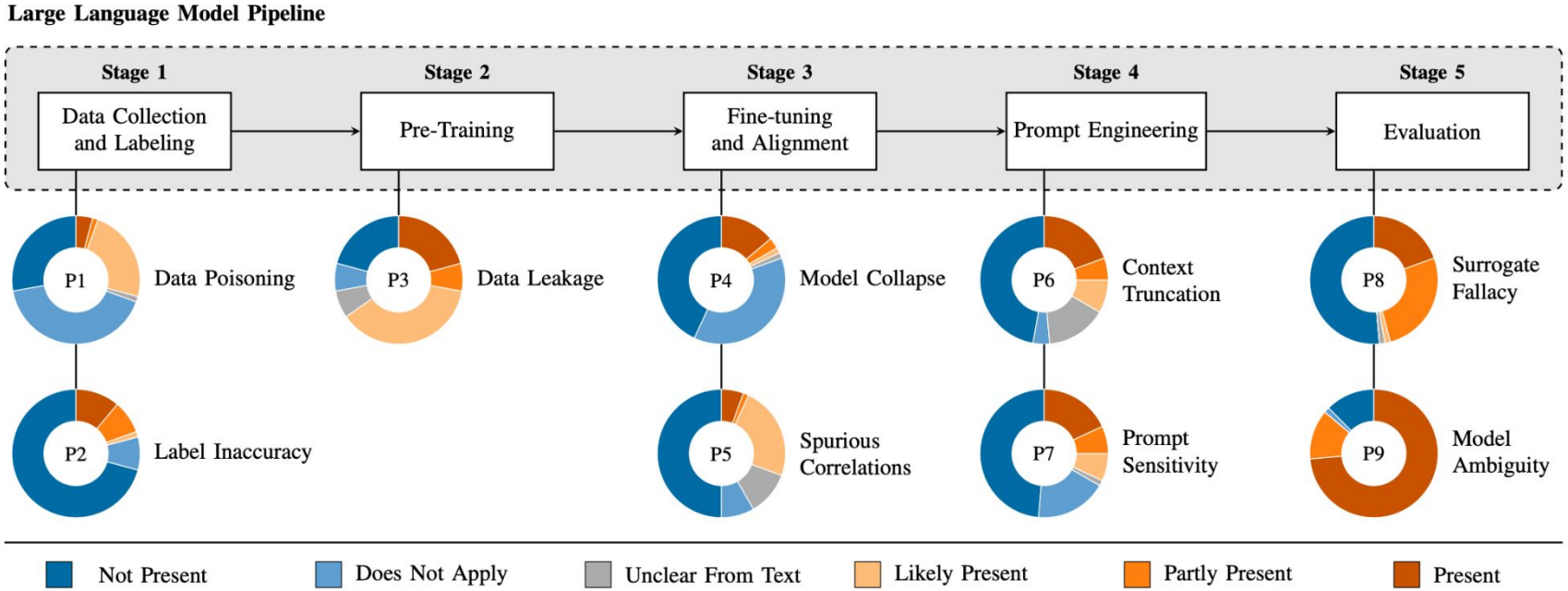


Fig. 1: Typical LLM pipeline, divided into its key stages. Each stage can introduce LLM-specific pitfalls that may distort evaluation, inflate reported performance, or undermine reproducibility. Colors indicate the prevalence of each pitfall according to our prevalence study (§III).

Nine Pitfalls Across the LLM Pipeline

P1
Data Poisoning

Scraped training data may hide planted content

P2
Label Inaccuracy

LLM-made labels used without checking

P3
Data Leakage

Test data may already be in the training set

P4
Model Collapse

Training on LLM output degrades the model

P5
Spurious Correlations

Model learns shortcuts unrelated to the task

P6
Context Truncation

Long inputs get cut at the context limit

P7
Prompt Sensitivity

Small prompt changes swing the results

P8
Surrogate Fallacy

Results from one model claimed for others

P9
Model Ambiguity

Exact model version or quantization not reported

72 LLM Security Papers from Top Venues

72

papers published Jan 2023 – Dec 2024

SECURITY

IEEE S&P · NDSS
ACM CCS · USENIX Security

SOFTWARE ENGINEERING

ICSE · FSE · ISSTA · ASE

INCLUSION CRITERION

LLM is integral to the core contribution

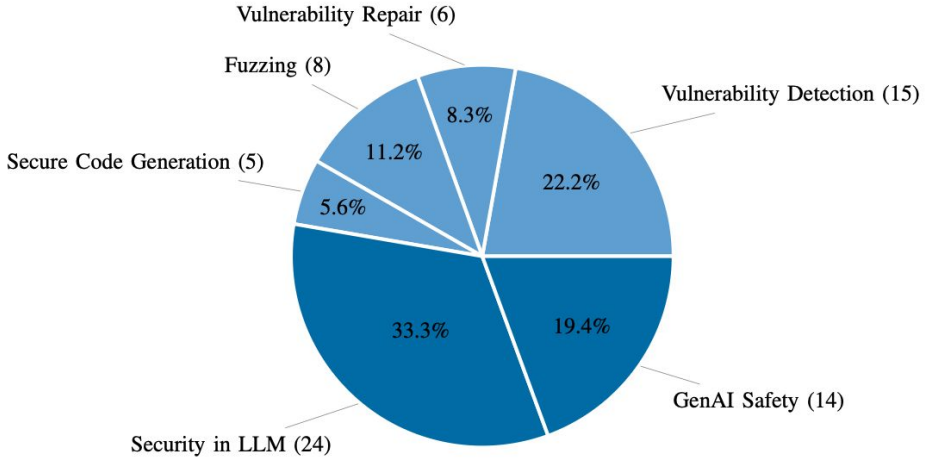
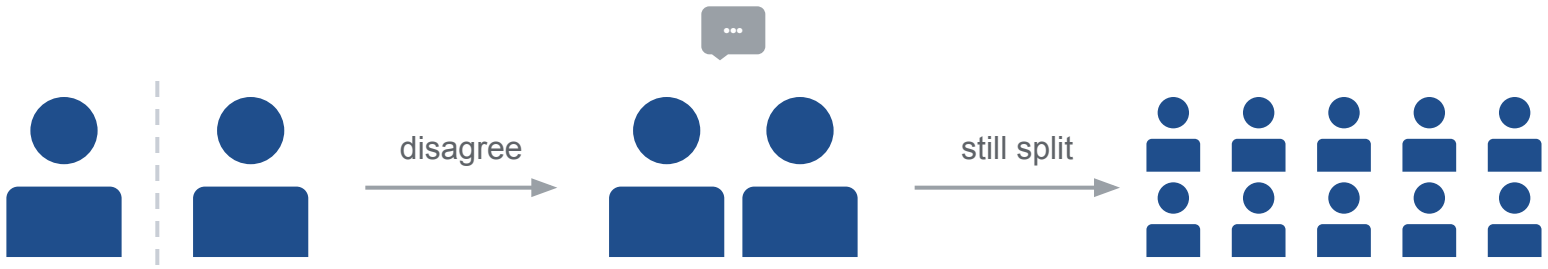


Fig. 2: Distribution of the 72 selected papers across topics. The top half, light blue segments (●) represent research that uses LLMs to address security problems. The bottom half, dark blue segments (●) correspond to research focused on the security and safety of LLMs themselves.

Review Process: 15 Reviewers, 3 Phases



PHASE 1

Blind review

2 reviewers per paper

PHASE 2

1:1 discussion

the same pair

PHASE 3

Group discussion

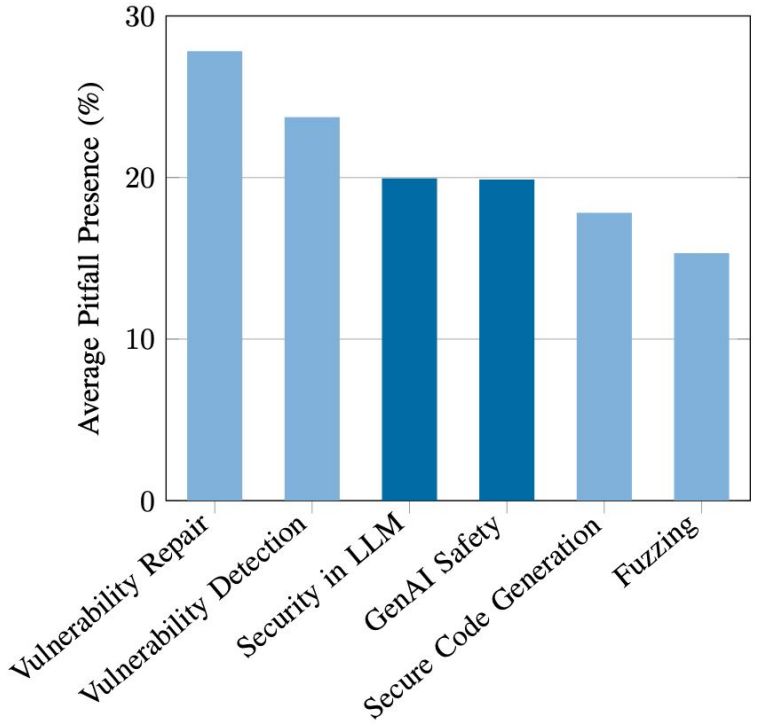
10 of 15 reviewers

Reviewers from security, ML, and software engineering; labeling guidelines refined in a pilot review

Every Paper Had at Least One Pitfall

Pitfall	Present	Partly present	Likely present	Total affected	Discussed
P9 Model Ambiguity	**53 (73.6%)**	9 (12.5%)	0	**62 (86.1%)**	**0% (0)**
P3 Data Leakage	15 (20.8%)	5 (6.9%)	**27 (37.5%)**	**47 (65.2%)**	21.3% (10)
P8 Surrogate Fallacy	14 (19.4%)	**19 (26.4%)**	1 (1.4%)	**34 (47.2%)**	14.7% (5)
P6 Context Truncation	14 (19.4%)	4 (5.5%)	6 (8.3%)	**24 (33.3%)**	41.7% (10)
P7 Prompt Sensitivity	13 (18.1%)	5 (6.9%)	5 (6.9%)	**23 (31.9%)**	21.7% (5)
P5 Spurious Correlations	4 (5.6%)	1 (1.4%)	17 (23.6%)	**22 (30.6%)**	9.1% (2)
P1 Data Poisoning	3 (4.2%)	0	17 (23.6%)	**20 (27.8%)**	**0% (0)**
P2 Label Inaccuracy	8 (11.1%)*	6 (8.3%)	1 (1.4%)	**15 (20.8%)**	**60.0% (9)**
P4 Model Collapse	10 (13.9%)	0	0	**10 (13.9%)**	**0% (0)**

Vulnerability Papers Fell into the Most Pitfalls



27.8%

Vulnerability Repair

23.7%

Vulnerability Detection

15.3%

Fuzzing (lowest)

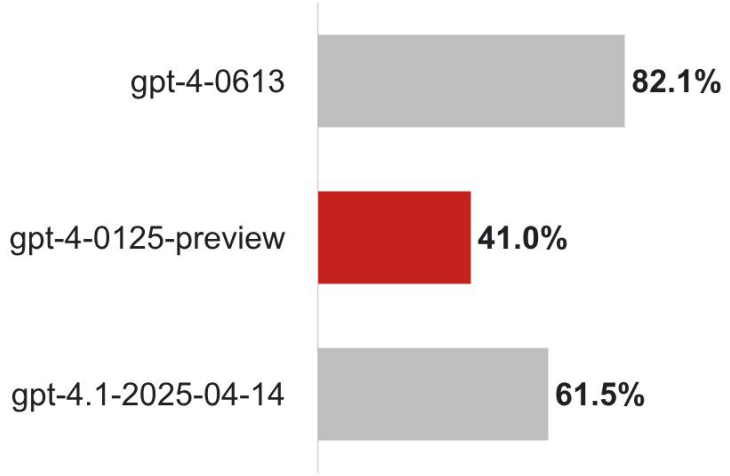
Average share of pitfalls present per paper

Same Model Name, Different Results

CASE STUDY 1 · P9 MODEL AMBIGUITY + P8 SURROGATE FALLACY

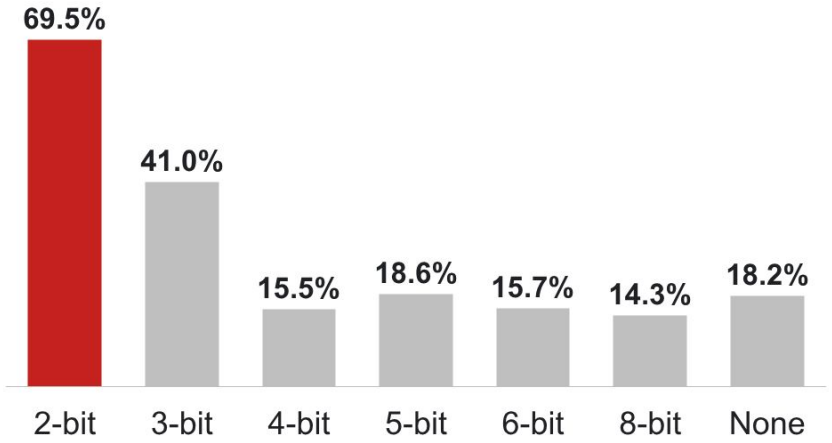
Hate speech detection recall

One method, three snapshots all called “GPT-4”



Jailbreak success rate, CodeLlama 7b

Ollama build, by quantization level



Source: Evertz et al., NDSS 2026, Tables I–II. Hate speech task re-implements an IEEE S&P 2024 method.

Leaked Test Data Inflates F1 Almost Linearly

CASE STUDY 2 · P3 DATA LEAKAGE

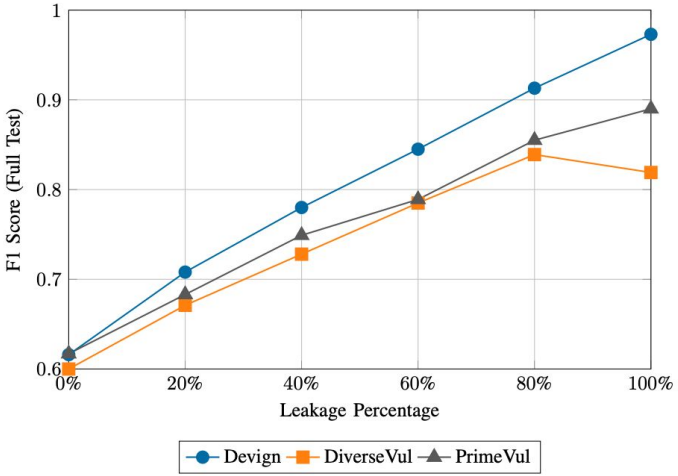


Fig. 4: F1-scores of CodeT5+ models fine-tuned on *Devign*, *DiverseVul*, and *PrimeVul* with varying amounts of test data (0–100%) leaked into the training data.

Source: Evertz et al., NDSS 2026, Fig. 4.

+0.08–0.11

F1 gain from leaking 20% of the test set

0.60–0.62

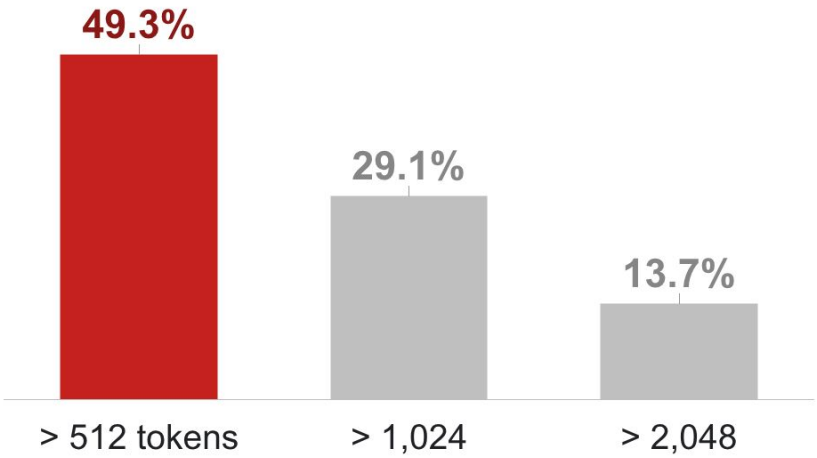
F1 with no leakage

CodeT5+ fine-tuned on Devign, DiverseVul, PrimeVul (60/20/20 split), with 20–100% of the test set added to training

Half of Vulnerable Functions Exceed 512 Tokens

CASE STUDY 3 · P6 CONTEXT TRUNCATION

Functions longer than the context window
Vulnerable functions, all 3 datasets combined



8 of 15

vulnerability detection papers used a model with a context of 512 tokens or less

CVE-2014-2669
The integer-overflow line sits beyond token 512, so a 512-token model never sees it

Over 512 tokens by dataset:
Devign 41.7% · DiverseVul 51.8% · PrimeVul 64.9%

Source: Evertz et al., NDSS 2026, Table IV. Lengths measured with the CodeT5 tokenizer.

Models Trained on Their Own Code Degrade

CASE STUDY 4 · P4 MODEL COLLAPSE

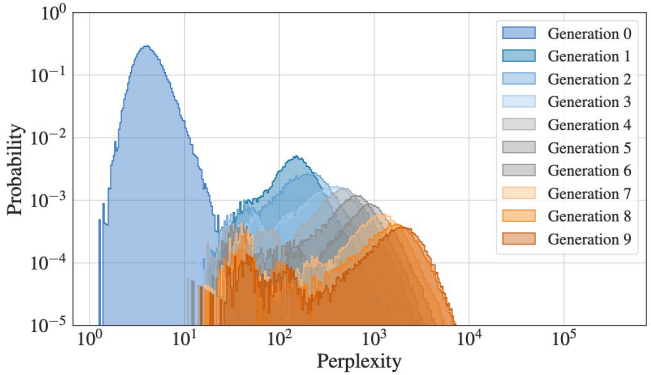


Fig. 5: Distribution of perplexities for training samples generated across multiple model generations. Perplexity is measured using the original model fine-tuned on real data. Both the mean and variance of perplexity increase over generations, suggesting that models become less stable when repeatedly trained on synthetic data.

Gen 0 → **Gen 1** → … → **Gen 9**

Each model fine-tunes on code from the one before

Low perplexity = looks like normal code
High perplexity = looks unusual

Each generation drifts further right

Errors pile up and outputs become unstable

Qwen2.5-Coder-0.5B, 50k Python samples, 10 generations; perplexity scored by the Gen 0 model

Source: Evertz et al., NDSS 2026, Fig. 5. Setup follows Shumailov et al., Nature 2024.

Recommendations

R1

Transparency and Reproducibility

P9 Pin exact model versions

P2 Validate LLM-made labels

P8 Test before generalizing

R2

Understand and Communicate Data Quality

P4 Disclose synthetic data

P1 Assess poisoning risk

P3 Use post-cutoff test data

R3

Alignment Between Models and Tasks

P6 Measure truncation

P7 Test prompt variations

P5 Check for shortcuts

Source: Evertz et al., NDSS 2026, §V. Living guidelines at llmpitfalls.org.

Limitations and Personal Thoughts

L1 **Reviewer team is barely described**

- Described only as 15 researchers in 3 fields
- No seniority, field split, or LLM experience
- Inter-reviewer κ is only **0.55** (moderate)

L2 **Surveyed papers are two years old**

- Papers from Jan 2023 to Dec 2024 only
- Nine pitfalls never mention agents or tool use
- Case studies test older models, e.g. GPT-4-0613

PERSONAL THOUGHT

A checklist for my own next paper

- Report exact snapshot and quantization
- Measure how often inputs are truncated
- Check test data against training cutoff
- Re-run this survey on 2025–26 papers, including agents