

**Reichman University**

**Prompt Impact on LLM Accuracy for Cyber-Security Tasks**

**Natural Language Processing - Final project 2025**

Dr. Kfir Bar

Mr. David German

Mr. kai Golan Hashiloni

Mr. Amir Cohen

Created by:

Harel Zalman, 208200972

**Introduction**

Small security teams frequently rely on off‑the‑shelf large language models to triage alerts, label URLs, or sift log streams. Yet they seldom have expert‑annotated datasets—or the GPU budget—needed for extensive fine‑tuning. Our goal is to quantify, with a single clear metric per task, **how much intelligently‑crafted prompts can elevate raw LLM accuracy**. Specifically, we evaluate three public datasets—(1) URL phishing versus legitimate sites, (2) Telegram spam versus ham messages, and (3) Twitter suspicious versus non‑suspicious tweets—to ground the comparison. We measure each task under four distinct configurations:

* **Zero prompt** – the bare model, exactly as shipped, “Reply with one token: ‘Phishing’ or ‘Benign’.
* **Generic security prompt** – a short, reusable instruction such as “You are a cyber‑security analyst. Answer concisely.”
* **Task‑specific prompt** – a bespoke instruction engineered for the dataset, e.g. "Think step‑by‑step internally. Look for special keywords...”
* **Lightweight fine‑tune** – a three‑epoch LoRA adaptation using the very same data split, representing the upper bound.

Practitioners can then decide if prompt engineering alone is “good enough” or if a targeted fine‑tune is worth the added complexity.

**Key Question**

How much can a well‑crafted task‑specific prompt narrow the accuracy gap to a lightweight fine‑tune across common cyber topics including phishing , spam detection, and suspicious messages?

**Related Work**

Phrasing alone can swing GPT‑3 accuracy by over thirty percentage points (*Liu et al., 2023*), and calibration tricks such as self‑consistency sharpen those gains (*Zhao et al., 2021*). In the security domain, most evidence remains anecdotal—red‑team prompt sheets, vendor blogs, or internal SOC playbooks. Instruction‑tuning increases robustness (*Wei et al., 2022*), but few studies place prompt‑only and tune‑only approaches on the same scoreboard under equal resource limits. Our contribution is the first cross‑task benchmark that does exactly that.

**Experimental Setup**

**Tasks & Datasets**

• **URL phishing** — PhishTank daily feed, labels: phish vs legitimate.  
• **Telegram spam** — Kaggle “Telegram Spam or Ham” dataset (<https://www.kaggle.com/datasets/mexwell/telegram-spam-or-ham>); messages labeled spam vs ham.  
• **Twitter suspicious** — Kaggle “Suspicious Tweets” dataset (<https://www.kaggle.com/datasets/syedabbasraza/suspicious-tweets>); tweets labeled suspicious vs non‑suspicious.

**Experimental Conditions**

1. **Baseline** — Llama‑3‑8B‑Instruct with no system prompt.
2. **Generic prompt** — reusable, domain‑neutral.
3. **Task‑specific prompt** — hand‑crafted for maximum clarity and brevity, using multiple approaches, heuristic and best‑practice applied at different context lengths.
4. **LoRA fine‑tune** — rank 8, α = 16, trained for three epochs with early‑stopping on dev F1; total VRAM < 12 GB.

**Metrics & Evaluation**

• Accuracy, precision, recall, and macro‑F1.  
• Accuracy change relative to the raw model.  
• End‑to‑end latency from prompt assembly to first token, averaged over 1 000 samples on both GPU and CPU‑only machines.  
All seeds, dataset splits, and checkpoints are fixed and logged for reproducibility.

**Implementation Notes**

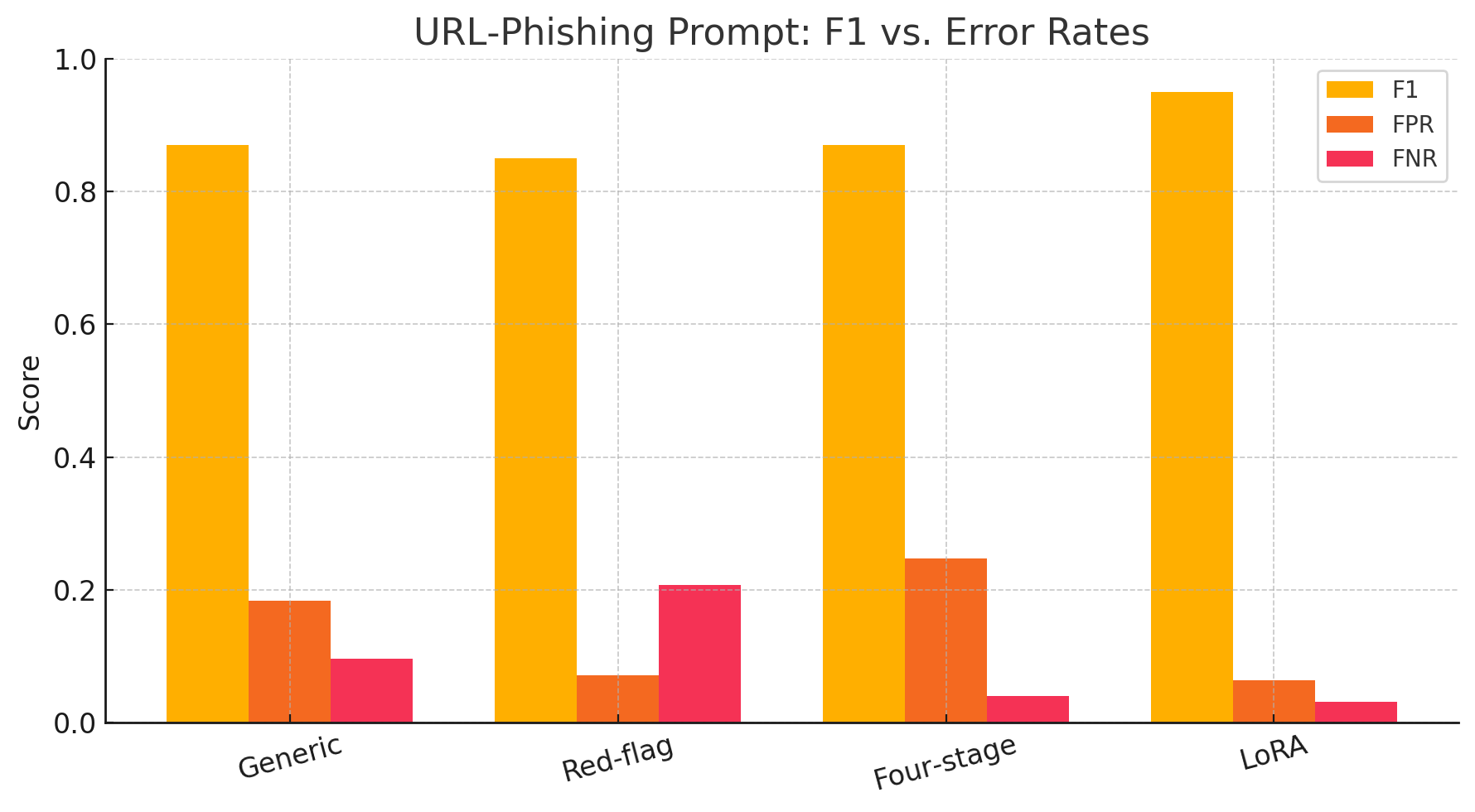
Evaluation relies solely on HuggingFace transformers 4.41 and llama‑cpp‑python 0.3.8, allowing fully offline runs. Fine‑tunes use PEFT LoRA; extra parameters weigh only 90 MB on disk. The harness (src/eval.py) streams each sample, captures logits and decisions, and writes JSONL for analysis.

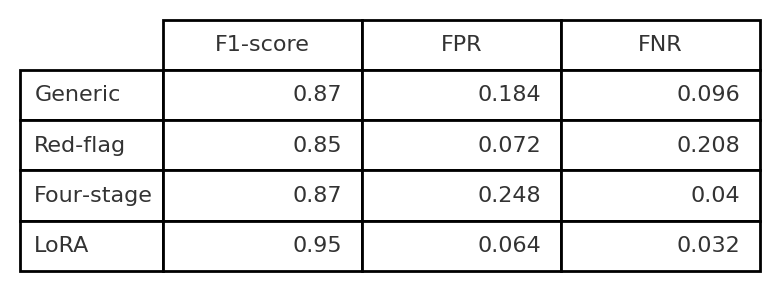
**Results & Analysis**

**URL Phishing Results**

We compared five inference configurations on the 250 test samples, URL dataset:

* **No Prompt** – Without any guiding instruction the model instantly labels every URL as “Phishing,” offering no meaningful discrimination.
* **Generic Prompt** – “Decide whether the following URL is legitimate or phish.” Precision 0.83, Recall 0.90, F1 0.87, False-positive rate 0.184, False-negative rate 0.096
* **Task‑Specific Prompt (Red‑flag criteria)** – “Decide whether the following URL is legitimate or phish. If you detect at least two phishing red flags (e.g., punycode, IP address host, credential keywords), mark it as phish; otherwise treat it as legitimate.” Precision 0.92, Recall 0.79, F1 0.85, False‑positive rate 0.072, False‑negative rate 0.208
* **Task‑Specific Prompt (Four‑stage analysis)** – “Analyse the URL in four stages—host, path, query, then overall—flagging IP hosts, high‑risk TLDs, suspicious keywords, and obfuscated encodings before reaching a verdict of legitimate or phish.” Precision 0.79, Recall 0.96, F1 0.87, False‑positive rate 0.248, False‑negative rate 0.04; notably only **5 false negatives**, the lowest FN count among all prompt variants.
* **LoRA Fine‑Tune** – model adapted with LoRA. Precision 0.94, Recall 0.97, F1 0.95, False‑positive rate 0.064, False‑negative rate 0.032.

****



**References**

1. Liu, P. et al. (2023). *Prompting GPT‑3 for Improved Few‑Shot Performance.*
2. Zhao, S. et al. (2021). *Calibrate Before Use: Improving Few‑Shot Performance of Language Models.*
3. Wei, J. et al. (2022). *Finetuned Language Models Are Better Instruction Followers.*