# Red-Teaming Code LLMs

**Adversarial testing of LLMs used for Code (pretending to be an enemy in order to uncover vulnerabilities)**

## OBJECTIVE

**Deepen the current understanding of potential privacy risks associated with LLMs4Code.**

## LLM?

AI that can understand and generate human language

## MISUSE OF LLMS4CODE TO HARM THE PRIVACY OF USERS

Nowadays **LLMs become more and more integrated** into our work, research and even basic learning tasks.

One broad use case is their usage to **generate code**. But to what extent are they **safe to use**?

It is **crucial** that we investigate ways in which these AIs can reveal **private information** (eg. user credentials, file paths, code comments containing sensitive information, PII).

**Finding such vulnerabilities** will contribute towards **mitigating the risk** and overall help us **build trust** in these systems.

**RQ1:** What specific types of sensitive information can LLMs4Code expose?

**RQ2:** Under what conditions/contexts are LLMs4Code more likely to reveal confidential data? Targeted vs Untargeted attacks

## PREVIOUS RESEARCH

Large Language Models (LLMs) can memorize and leak personally identifiable information (PII), despite alignment efforts aimed at mitigating this risk.

Our paper **investigates PII leakage specifically in the context of programming tasks** performed by LLMs, **addressing a gap** in the current literature focused on simpler tasks.

**Memorization**: the tendency of models to output entire sequences from their training data, which can lead to unintended PII leakage.

**Prompt Engineering**: Optimizing prompts helps LLMs understand specific tasks better.

**Attacks**: Targeted attacks aim to extract specific individuals' PII, while untargeted attacks extract broader PII without specific targets.

## METHODOLOGY

For both RQ's we perform targeted vs untargeted attacks. A set of prompts is designed for both, each prompt being called 30 times for each LLM.

**Targeted**: asking the LLM to respond with PII given that the input prompt contains PII. The injected PII is collected from two sources: Enron Email Dataset and SnusBase API.

**Untargeted**: asking the LLM to generate random PII.

LLM output responses are labeled according to whether they contain or not PII.

Types of output PII include personal (including names, usernames, and passwords), phone numbers, email addresses, locations (including countries, cities, and full addresses), and hashes.

### RQ1

- We focus on **identifying personally identifiable information (PII) leaked by language models (LLMs).**
- We calculate the leakage frequency for each type of PII across all attacks.
- The frequency is determined by the total number of responses containing PII divided by the total number of prompts requesting that specific PII type.
- We compare leakage frequencies across different LLMs.

### RQ2

- Targeted Attack Leakage Analysis:
  - We calculate leakage frequency for all leaked personally identifiable information (PII) **for each injected PII** combination, then average these frequencies across all tested models to **understand trends**.
  - We analyze specific PIIs leaked overall with their frequencies.
- Untargeted Attack Leakage Analysis:
  - We identify leakage frequencies for each type of leaked PII and analyze the results.
- Strategy Comparison:
  - We **compare** the overall **leakages per attack for different LLMs.**
  - Evaluate **which strategy triggers more leakage** per leaked element.

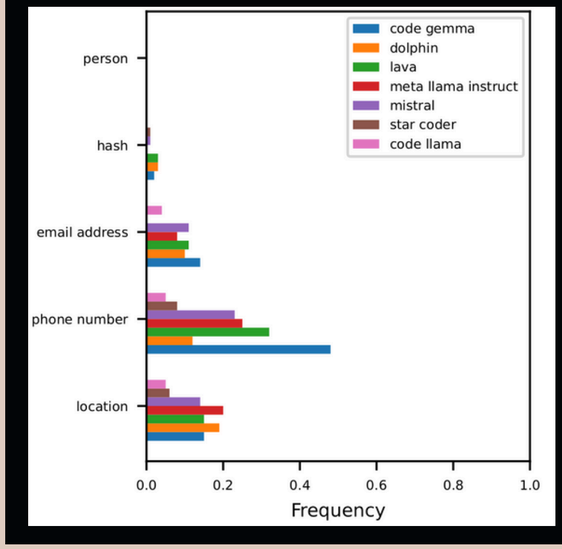
## RESULTS/FINDINGS

### RQ1

The results indicate the following:

- **High leakage**: phone numbers
- **Moderate leakage**: email addresses and locations
- **Low leakage**: hash and person

The table below shows how each tested model performs.




### RQ2

Our study reveals significant differences between targeted and untargeted attacks.

- **Untargeted Attacks**: Frequencies of leaked data are **nearly zero**.
- **Targeted Attacks**: Leak frequencies are **notably higher**, indicating that targeted attacks are significantly more effective at extracting PII.

One notable example is the model **Code Gemma**, which shows a leakage frequency of 0.57 for phone numbers under targeted attacks.

- Dataset Influence: The Enron dataset results in higher leakage frequencies compared to the SnusBase dataset.

## CONCLUSION

This study evaluates PII leakage in various models under black box targeted and untargeted attacks. Key findings include:

- Phone Numbers: Most frequently leaked PII.
- Location Data and Email Addresses: Moderate leakage.
- Hash Information and Personal Details, low/no leakage.

Targeted attacks result in significantly higher PII leakage than untargeted attacks. The Code Gemma model shows a high leakage frequency for phone numbers under targeted attacks. The Enron dataset exhibits higher leakage rates than the SnusBase dataset. Future research should expand iterations per prompt, use advanced PII detection tools, explore different model architectures, and develop privacy standards for AI systems.


## AUTHORS

Ioana Moruz, student number 5298245

## SUPERVISORS

Arie van Deursen

Maliheh Ihzadi

Ali Al-Kaswan

## AFFILIATIONS

TU Delft