Data Leaks Through Speculative Decoding: Network Side-Channel Attacks on LLM Inference

Sachin Thakrar, Yale College (B.S. Computer Science)
Advisor: Timothy Barron, Department of Computer Science, Yale University

Problem & Motivation

Large Language Models (**LLMs**) are everywhere. We increasingly use them for a variety of tasks in sensitive areas like healthcare, finance, and education. However, LLMs can also be slow and costly to serve, and so techniques like **speculative decoding** are utilised to speed inference up.

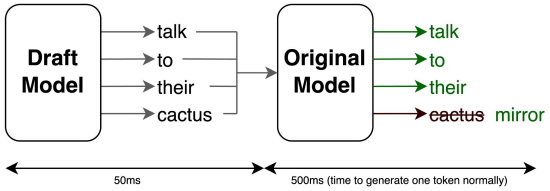
Most LLMs **stream** the response token-by-token to the user over encrypted channels. However, **subtle differences** in packet timings and sizes—caused by optimisations like speculative decoding—can **leak private user information**. A passive observer can exploit this to infer sensitive attributes about encrypted data.

We show the viability of this attack at a large-scale, on state-of-the-art LLMs from a variety of providers (OpenAI, DeepSeek, Llama).

Speculative Decoding Explained

The trick in speculative decoding is to get a small (draft) model to guess the next few tokens. We then feed the guessed tokens to the true model, which can **verify** those tokens in **one forward pass**, no matter the number of tokens—i.e., the time to generate one token! We can keep all the tokens up to where the draft model made a mistake.

User prompt: What do people talk to when alone?
Current Response: People sometimes....



When we stream these tokens back to the user, we can figure out which tokens were generated by the draft model and which weren't. This forms a **unique pattern** that serves as a side-channel. Certain queries, topics, and ideas will alter that pattern, allowing us to derive information about the query. For example, harder topics might result in more failures.

KEY FINDINGS

> LLM optimisations like **speculative decoding leak private user data** when streaming—encrypted packet timings reveal query attributes like medical conditions and user language.

> **Attackers require no direct server access**, just some knowledge of the prompt structure and which LLM is being used

> **High accuracy across diverse tasks**—achieved over 70% top-1 and 90% top-3 accuracy for identifying **medical diagnoses**, language, and educational topics

> By delaying tokens or padding packets, we can reduce efficacy significantly, with low cost.

Threat Model

Our attacker is a **passive network observer**, like a hacked ISP or browser extension. They can **see encrypted packets**, and their associated timings, size, and metadata.

They also know the **LLM used**, the **prompt format**, and the **possible categories**. This allows them to build a training set. It also isn't an unreasonable assumption—fixed prompt formats are increasingly common.

Experimental Setup

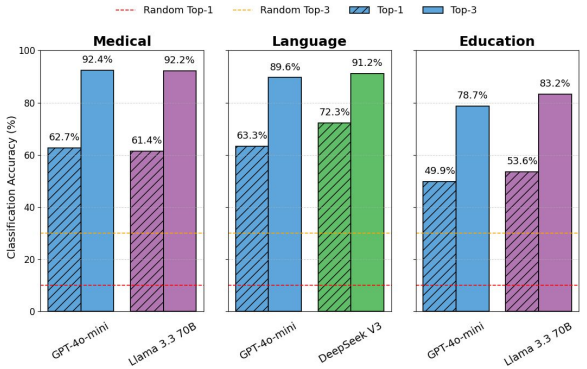
We target **DeepSeek V3 (0324)**, **GPT-4o (mini)**, and **LLaMa 3.3 70B Instruct**. We work on three unique attacks:

1. **Medical:** Can we determine the diagnosis of a patient, with a list of symptoms/issues as input?
2. **Languages:** Can we extract the query language?
3. **Topic:** Assuming some kind of educational (high-school / college level) question, can we extract the topic?

In each scenario, we use **ten categories**, and our adversary trains on 1000-1200 queries per category. They extract relevant features—like inter-packet delays, round segmentation, and inter-token delays that reflect the unique pattern, and train a classifier model on their data.

The victim is then simulated using a different VM, and we use the previously trained classifier on those network outputs.

Results



As seen above, we achieve **accuracies** between **50-70%**, depending on the task and model. Top-3 accuracies are consistently higher, too, indicating that the model, when incorrect, often ranks the true answer highly!

In this case, it is because **most misclassifications are between similar classes**. In the medical task, three of the ten diagnoses are respiratory tract infections, and those are regularly misclassified between each other. In the language task, the model mixes up French and Spanish, both of which are Romance languages.

Takeaways & Mitigations

LLM inference optimisations like speculative decoding can result in significant **privacy risks**, as passive network observers can infer attributes from timing patterns alone.

However, there are potential **mitigations**. As we rely on being able to distinguish between speculative and non-speculative tokens and their timings, we can:

- Hide timing data by sending tokens at fixed intervals
- Hide the number of tokens in a packet by padding each packet to a fixed size
- Send a fixed number of tokens each packet

All of these will significantly reduce the attack efficacy. Google now sends a fixed number of tokens (64) per packet, massively reducing attack efficacy.