Bit-by-Bit : The Side Channel Hackathon

Hackathon Challenge 2

ModelSpy: Identify the CNN model from Side-Channel CPU Traces

September 21, 2025

Background

Cloud hosts often partition powerful CPUs between authenticated users. Now, suppose you are a subscribed user of Amazon AWS using CPU resources for your work. One late evening after completing your work, you are sitting in your terminal casually, watching the system-level telemetry using perf, just routine monitoring of your own partition. While sipping your coffee, you observe a familiar pattern flickering across the charts: the cadence of cache misses, instruction cycles, and branch events looks strikingly like the inference run of a CNN model you've trained and profiled before. It is highly probable that, across the resource fence, another user is quietly running a model **X** for their own purpose, unaware that subtle side-channel fingerprints of their workload leak into the partition-level performance counters that you can observe.

The Challenge

The above story sets the stage for a side-channel analysis challenge, where you have labelled perf traces for a set of candidate CNNs and a fresh unlabelled capture from the shared CPU. Can you, using only the CPU's side-channel telemetry, identify whether the co-tenant's model inference corresponds to your known CNN model **X**? Your task is to build an algorithm that can identify which CNN model from the reference set the victim is running.

Your solution will be judged on accuracy, robustness (works across system noise and small timing shifts), efficiency (fast inference from traces), and explainability (why the classifier thinks a model matches).

What You Have

1. Possible options for target CNN models: **Resnet**, **AlexNet**, **VGG**, **DenseNet**, **Inception_V3**, **MobileNet_V2**, **ShuffleNet_V2**.
2. A small set of labeled target traces, captured from different runs/loads stored in **profiled_data** folder in **home** directory.

1. Trace Acquisition

Collect traces using Linux perf. Example commands participants can adapt:

```
perf stat -I 50 -e <event> -- /home/hackathon/dist/model_inference
```

Notes:

1. Replace event with platform-supported events.
2. Use multiple runs and warm ups; collect traces under realistic noise (other background jobs) to mimic cloud conditions.

Cloud Platform Credential:

- **IP**: `10.5.30.70`
- **UserID**: `<to be provided separately to each team>`
- **Password**: `<to be provided separately to each team>`

2. Evaluation Criteria

- **Primary**: Classification accuracy on hidden evaluation traces (weighted 60%).
- **Secondary**: Model explainability, lightweight runtime, and clarity of report (40%).

Tiebreakers: smaller model size and faster prediction time win.

3. Submission

To complete the challenge, please prepare the following:

- A brief report (PDF format) describing your methodology, analysis techniques, and any challenges you faced.
- Analysis file that contains final result.
- A link to a **public GitHub repository** containing all your scripts with report.

Judging & Feedback

Winners will receive detailed feedback from judges and a short highlight, asking them to present their methodology. The public leader-board will show accuracy only.

Good luck, and may the side channels be ever in your favor! Happy Hacking!