Deep Learning to Detect Novel Behaviours in Traces

Quentin Fournier

Polytechnique Montréal
Laboratoire DORSAL

Novel Behaviours

- Novel behaviours are deviations from what has been previously observed
  - Shift in normal behaviours: user patterns change over time
  - New normal behaviours: new users or features
  - Rare behaviours: unique queries
  - Anomalous behaviours: latency, attacks, or bugs

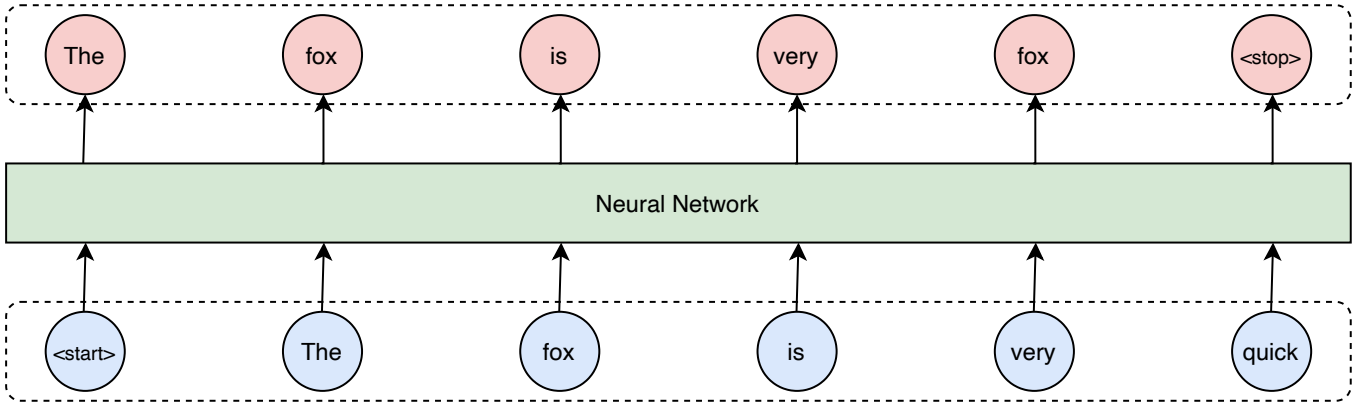
Approach - Dataset

- Sequences of system calls with their arguments
  - Expose the system behaviour
  - Easy to collect large datasets
  - Arguments enable more accurate predictions

Approach - Objective

- Learn an unsupervised language model
  - Labelling is time consuming and error-prone
  - Labels may change over time
  - Predict the next system call knowing the previous ones
  - Compute the likelihood of sequences

Approach - Objective



Perplexity

- The perplexity reflects how confused is the model about the input data
- The higher is the perplexity, the less likely is the input data under the model
- Set a threshold on the perplexity above which we consider the data to be unexpected and classify it as novel behaviour

Approach - Models

- Finite state machine *n*-gram
  - Conceptually simple and interpretable
  - Unable to model complex and long-range interactions
- Well known and proven neural network called LSTM
  - Designed to process variable-size sequences
  - Limited interaction range

Approach - Models

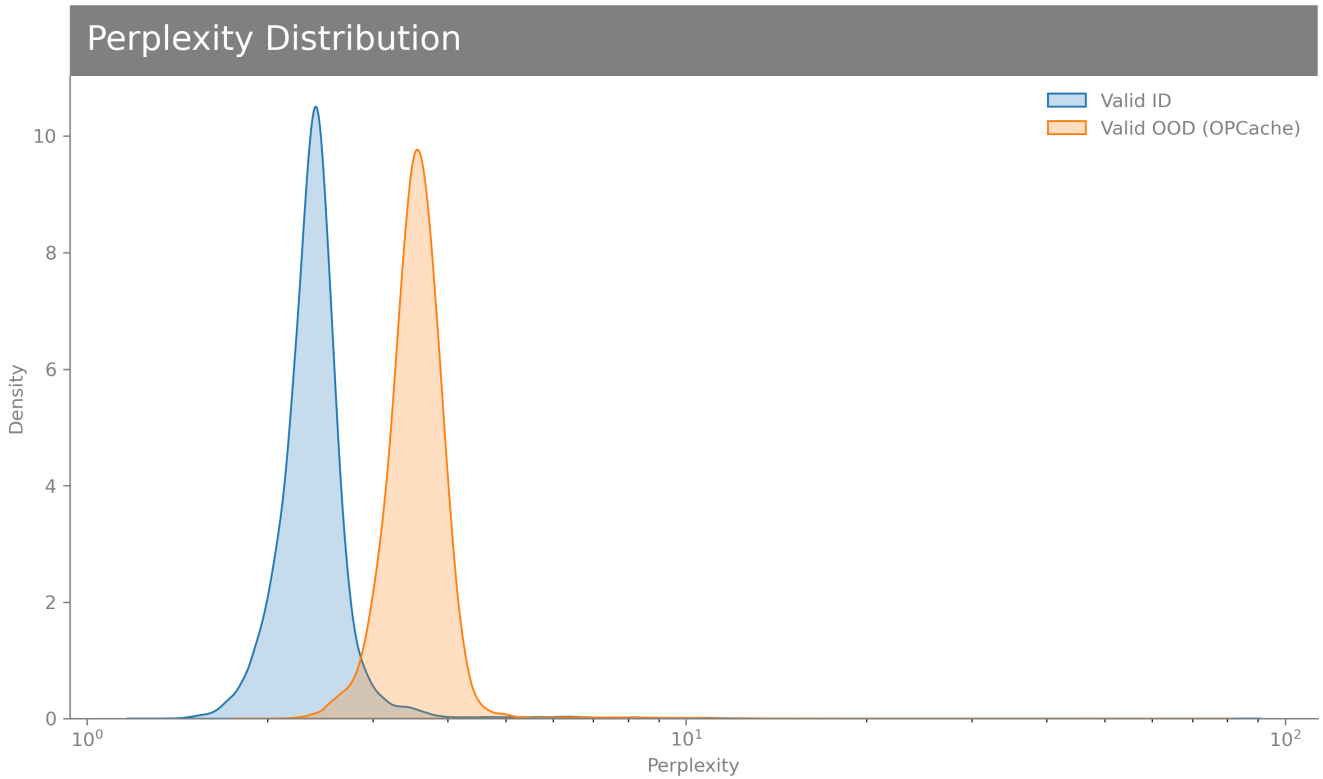
- State-of-the-art neural network called Transformer
  - Designed to model complex and long-range interactions
  - Memory intensive and sensitive to hyperparameters
- An efficient Transformer called Longformer
  - Able to model complex and long-range interactions
  - Significantly less memory intensive

Results - Datasets

- 1 known behaviour comprising 1,000,000+ web requests
- 6 novel behaviours comprising 100,000+ web requests each
  - Connection
  - CPU
  - IO
  - OPCache
  - Socket
  - SSL
- Simple and varied enough to evaluate the methodology
- Too simple to represent large-scale real-world use cases

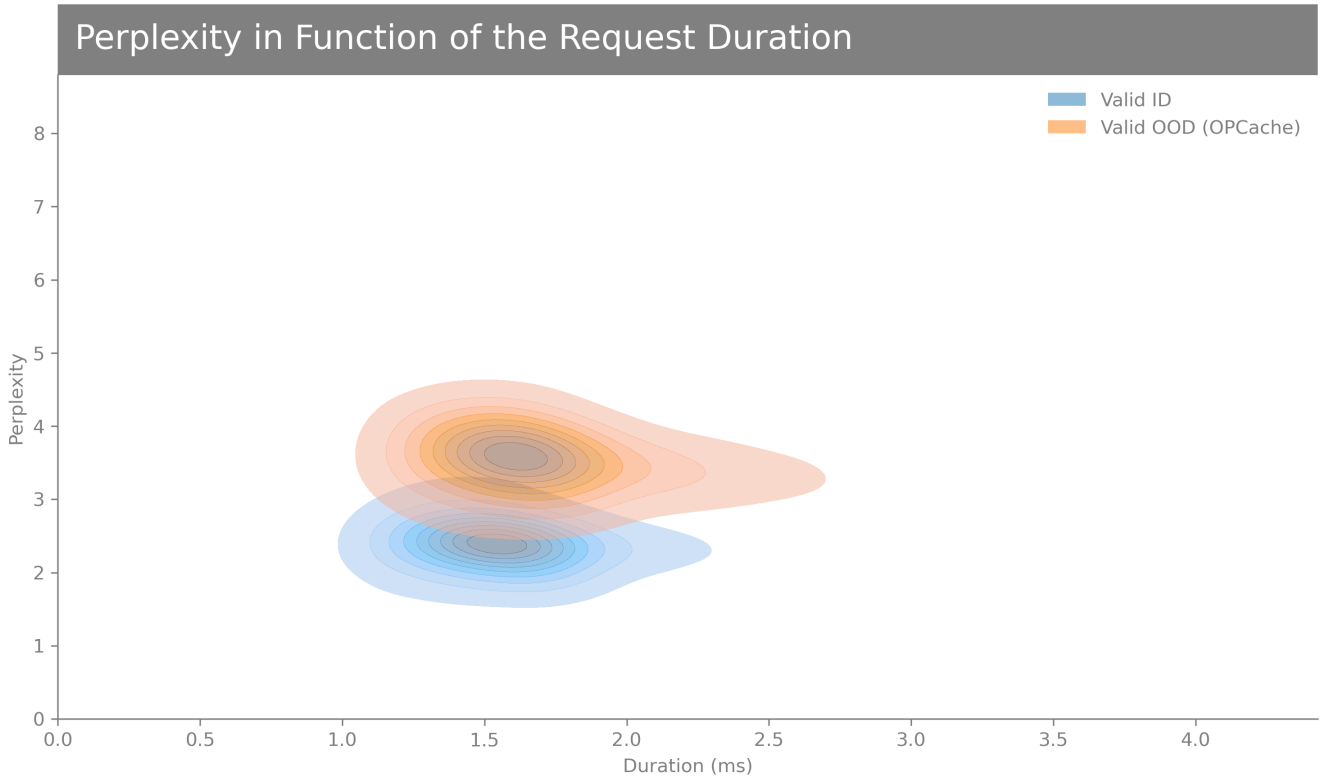
Results - LSTM - OPCache Disabled

- The novel behaviour has a higher perplexity than the known behaviour (small overlap)
- Accuracy of 95.56%



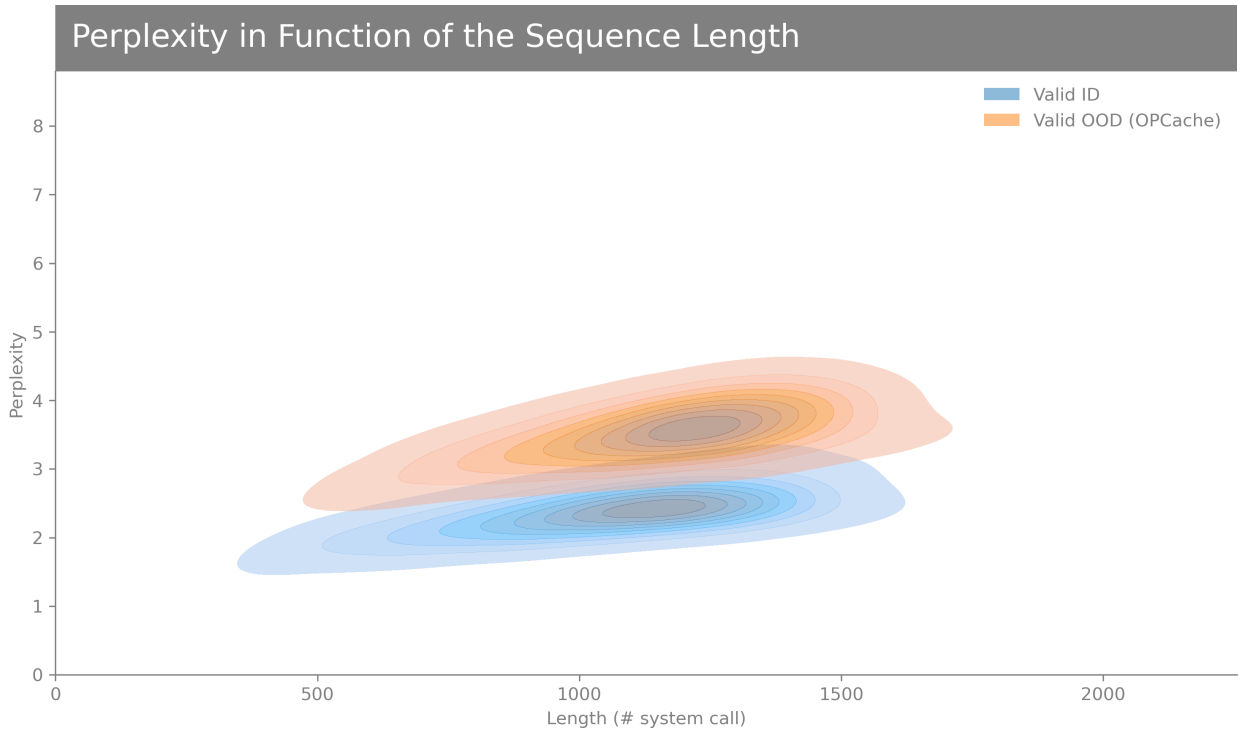
Results - LSTM - OPCache Disabled

- *x*-axis: known and novel behaviours have similar durations
- *y*-axis: the novel behaviour has a higher perplexity



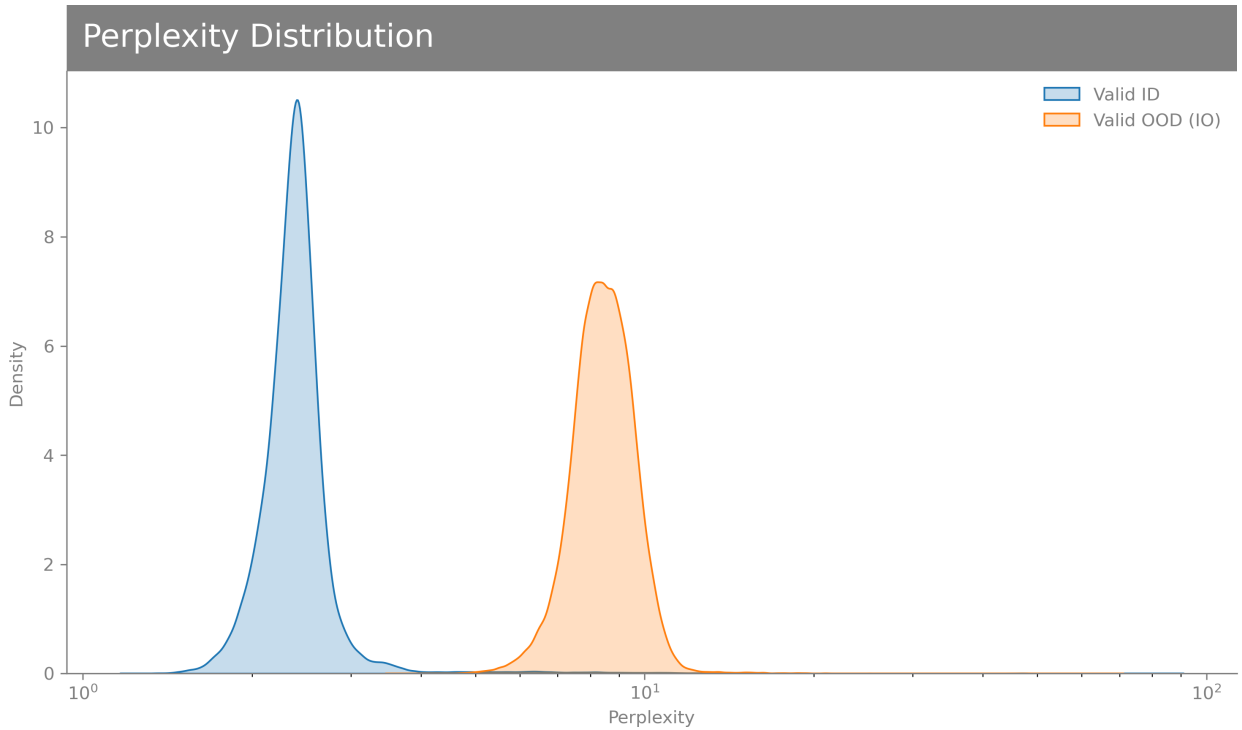
Results - LSTM - OPCache Disabled

- *x*-axis: known and novel behaviours have similar lengths
- *y*-axis: the novel behaviour has a higher perplexity



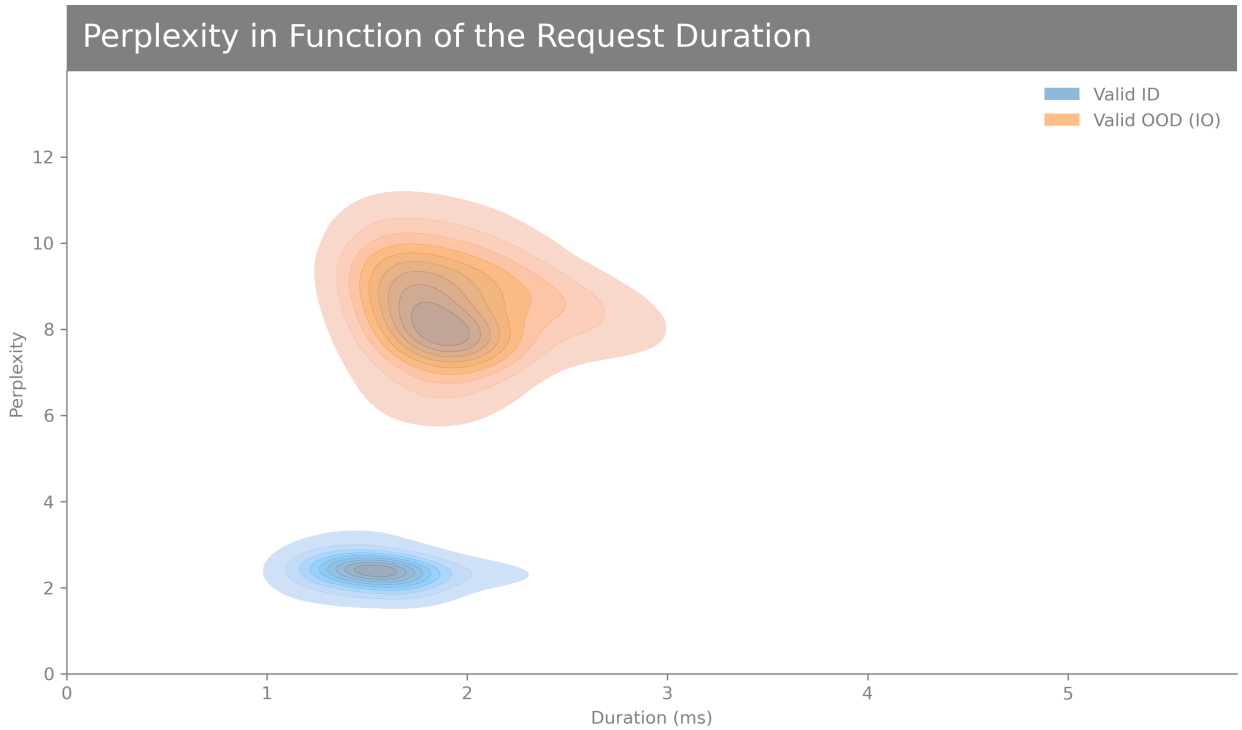
Results - LSTM - IO

- Best-case scenario, the novel behaviour has a significantly higher perplexity (no overlap)
- Accuracy 99.48%



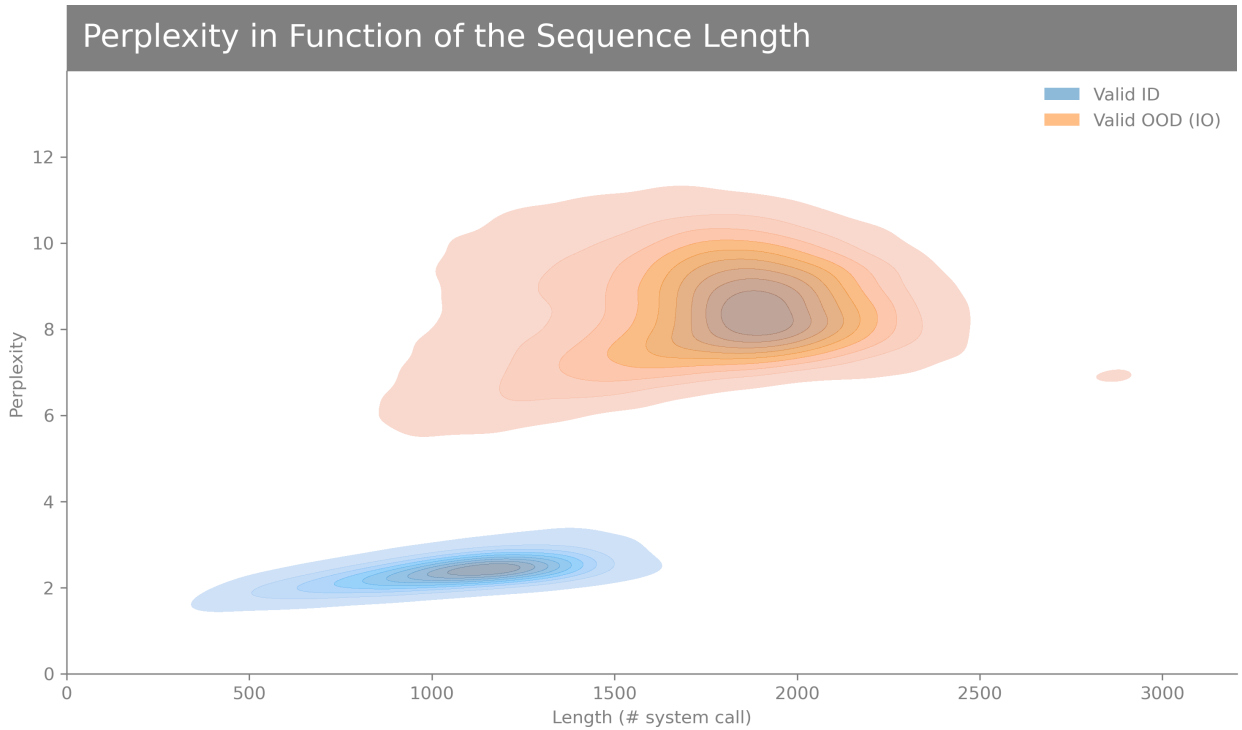
Results - LSTM - IO

- *x*-axis: known and novel behaviours has similar duration
- *y*-axis: the novel behaviour has a significantly higher perplexity (clear separation)



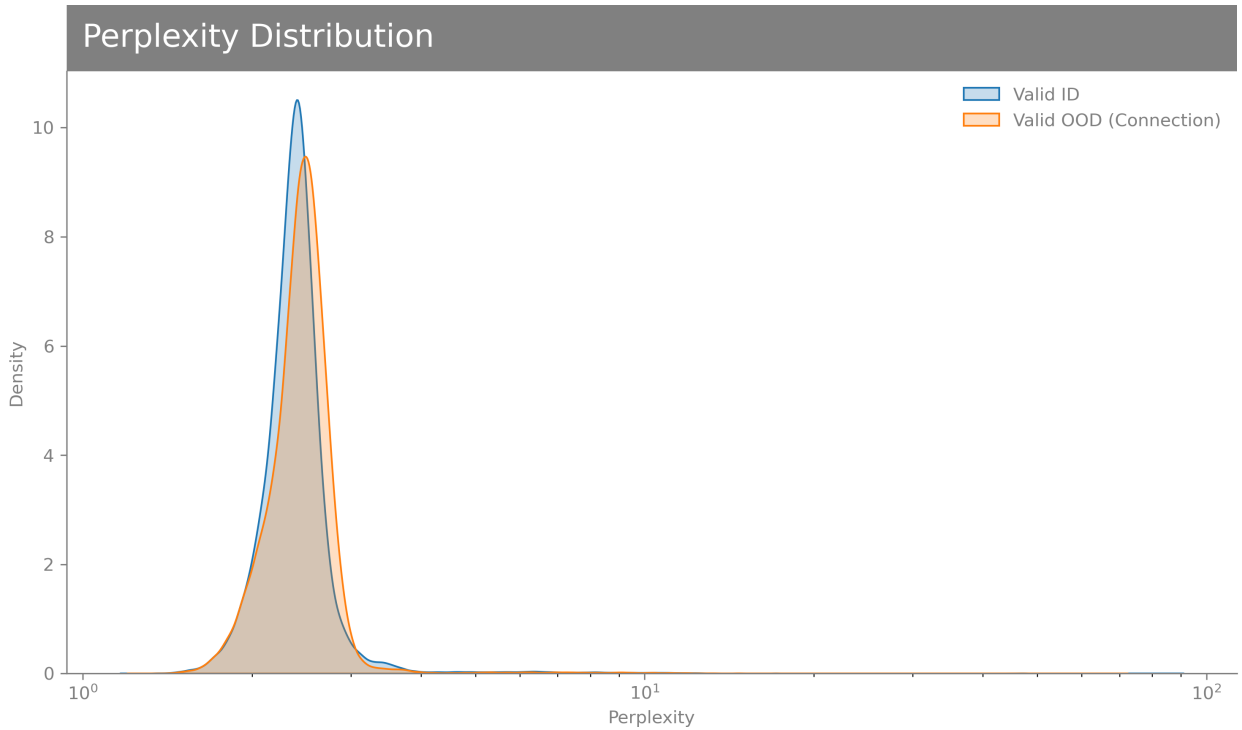
Results - LSTM - IO

- *x*-axis: the novel behaviour has significantly more system calls
- *y*-axis: the novel behaviour has a significantly higher perplexity (clear separation)



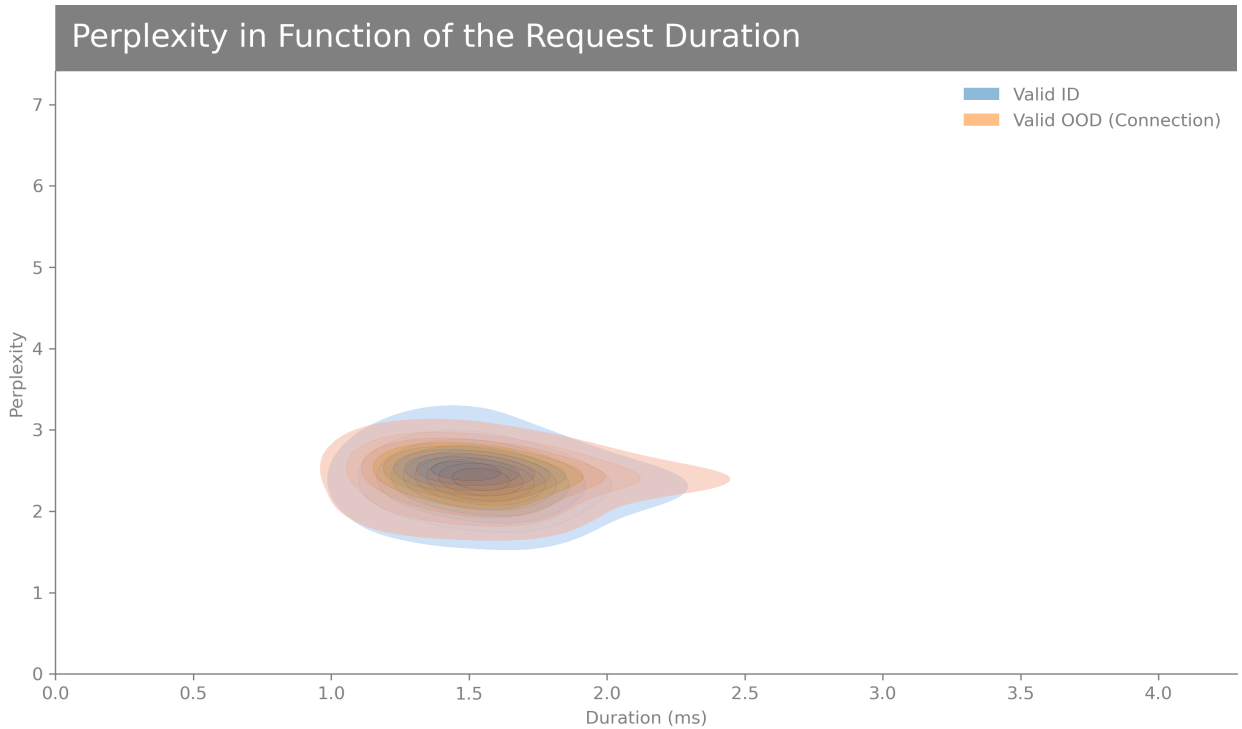
Results - LSTM - Connection

- Worst-case scenario, the novel behaviour has the same perplexity as the known behaviours (large overlap)
- Accuracy 50.23% (random)



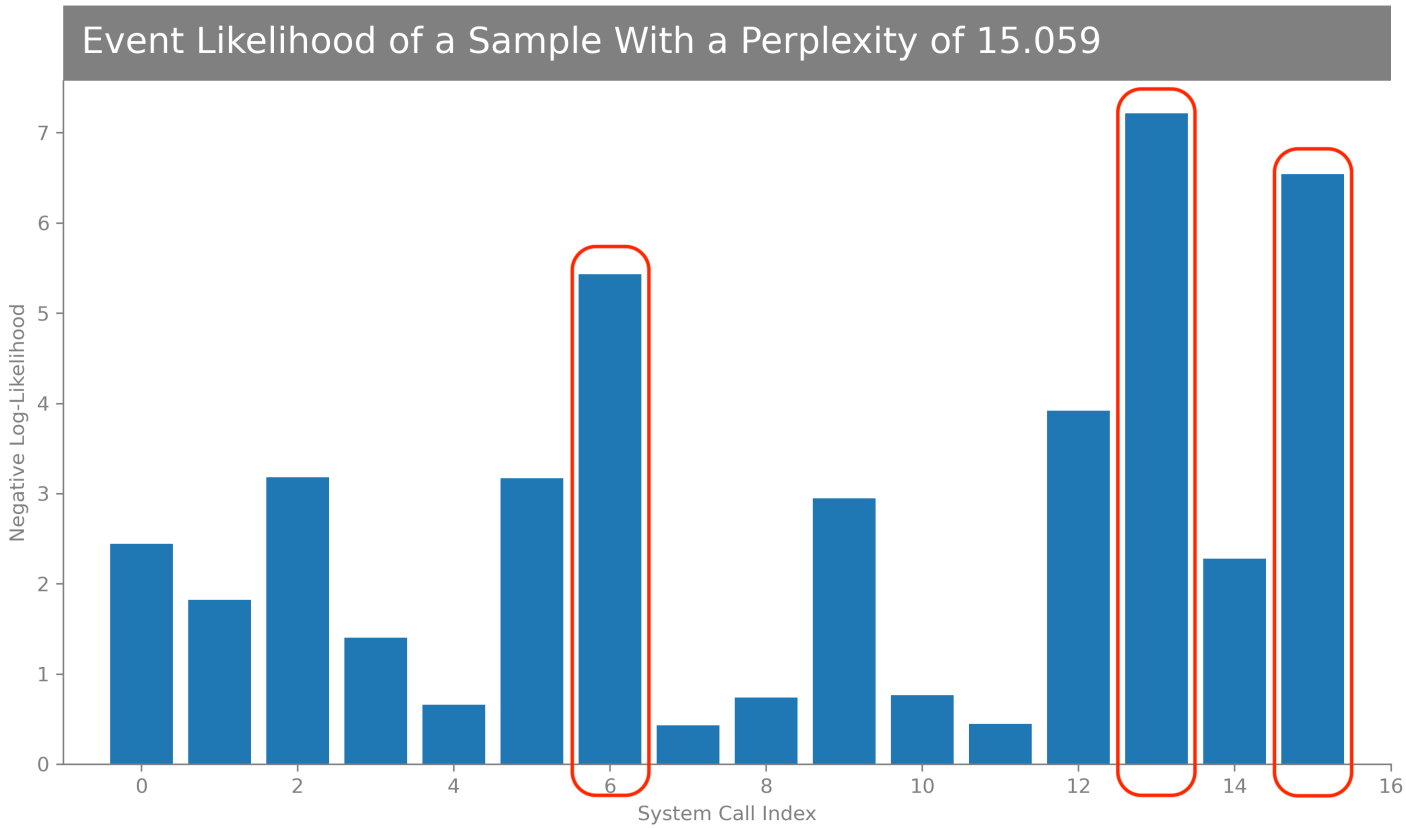
Results - LSTM - Connection

- The model is not able to separate between known and novel behaviours



Results - Root Cause Visualization

- What system calls are responsible for the perplexity?



I need you!

I need you!

- I am looking for a real-world use-cases:
  - Evaluate the proposed approach
  - Validate the synthetic results
  - Support the paper

I need you!

- I am looking for a real-world use-cases:
  - Train set of 10,000+ samples
  - Evaluation set of 100+ samples
  - Kernel or userspace traces with or without arguments
  - Samples length between 10 and 10,000 events

I need you!

- Applications: client, server, database, micro-services, etc.
- Sources: simulation, in production, deployed, etc.
- Behaviours: latency, bugs, attacks, software update, hardware changes, misconfigurations, etc.

Thank You