U.S. NAVAL RESEARCH LABORATORY

Repositioning Real-World Website Fingerprinting on Tor

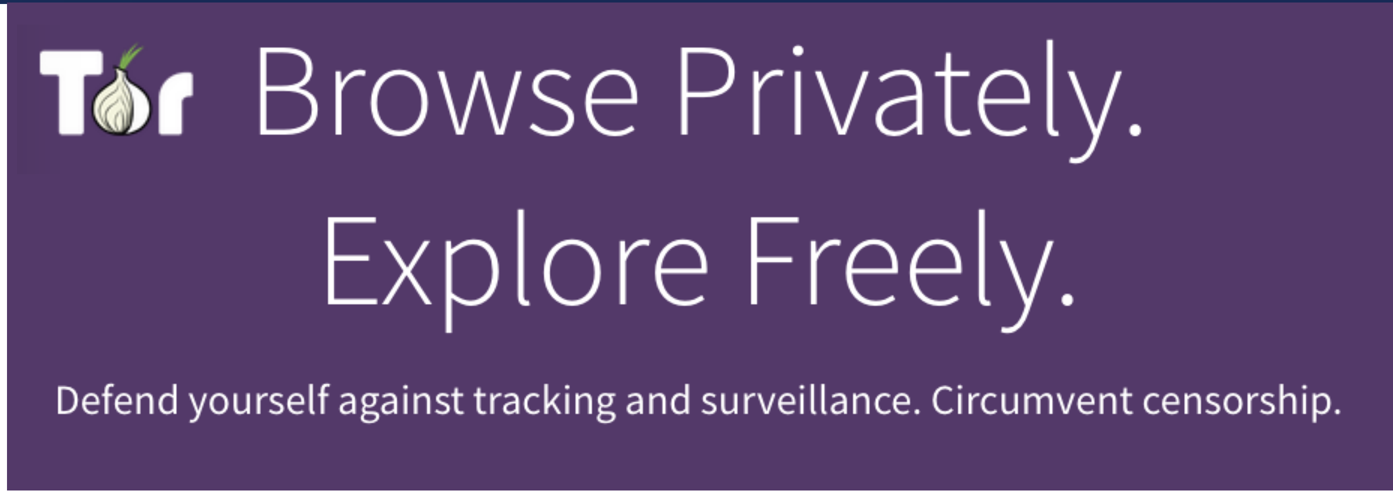
Rob Jansen, U.S. Naval Research Laboratory
Ryan Wails, U.S. Naval Research Laboratory and Georgetown University
Aaron Johnson, U.S. Naval Research Laboratory

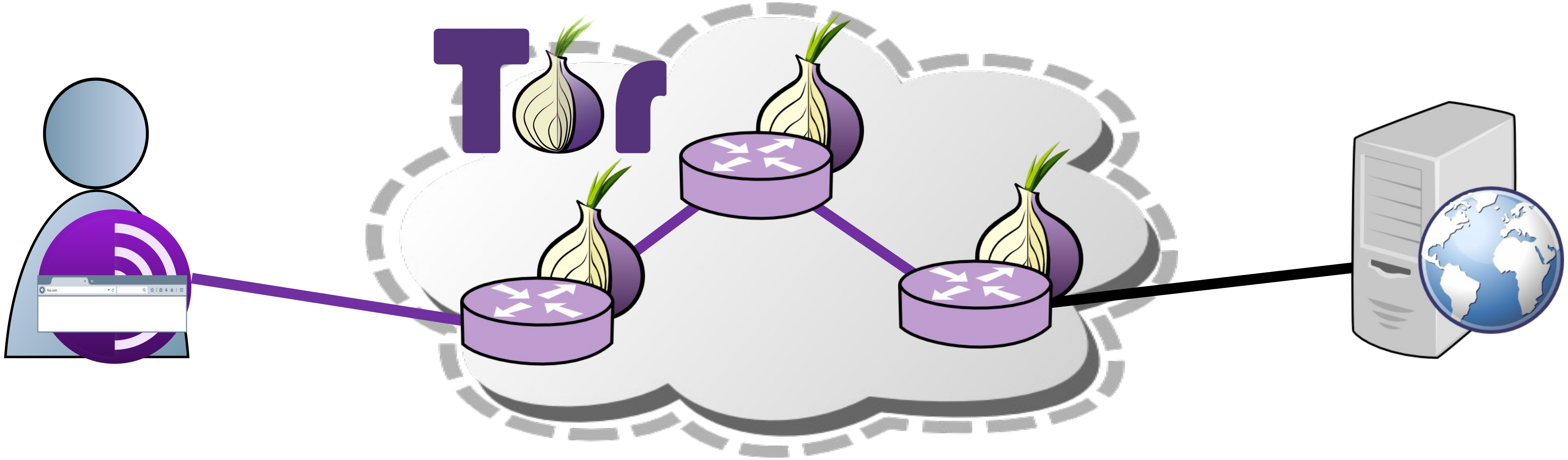
Rob Jansen, PhD
Computer Scientist
Center for High Assurance Computer Systems
U.S. Naval Research Laboratory

The Workshop on Privacy in the Electronic Society 2024
Salt Lake City, Utah, US
October 14th, 2024

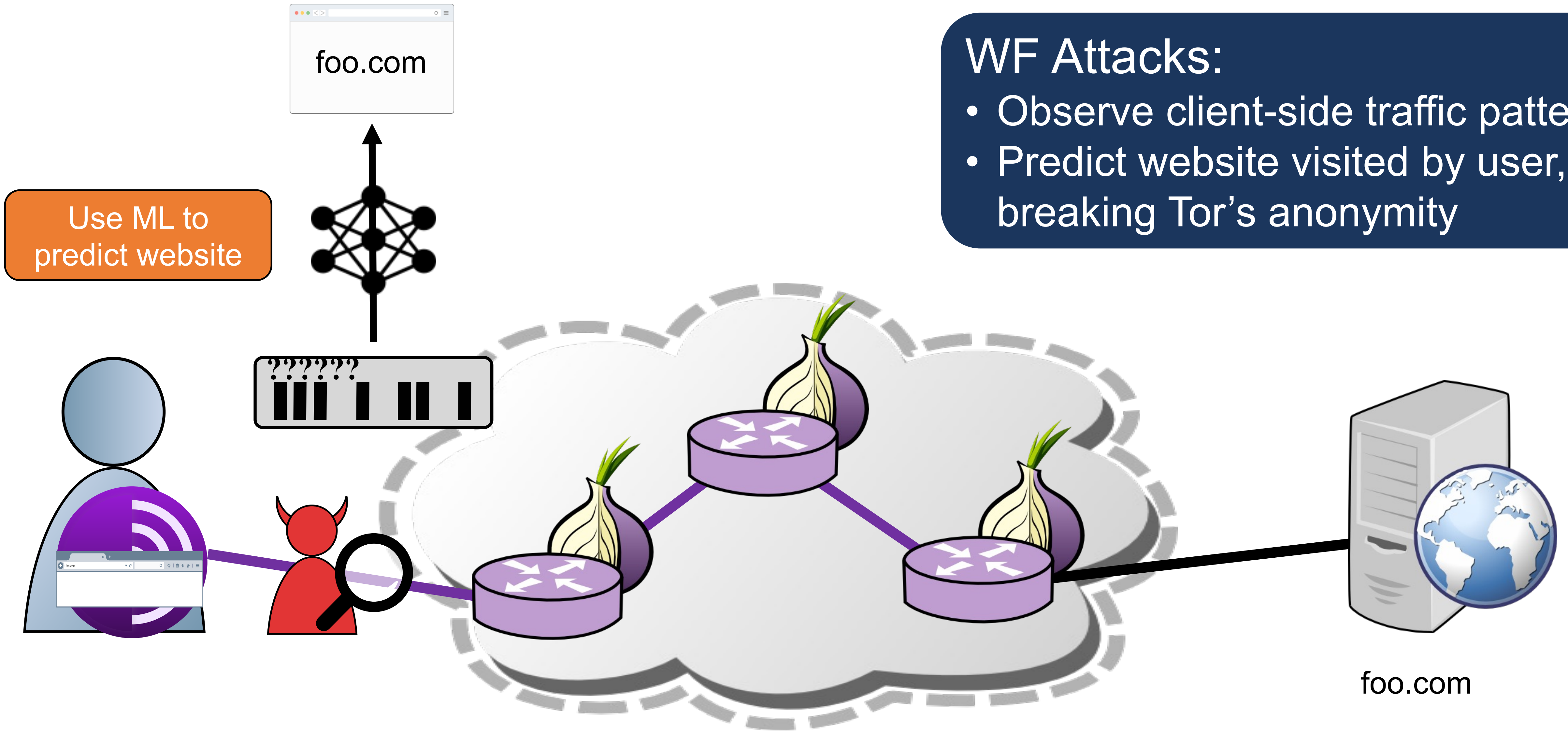
Anonymous Communication with Tor

- Separates *identification* from *routing*
- Provides unlinkable communication
- Promotes user safety and privacy online



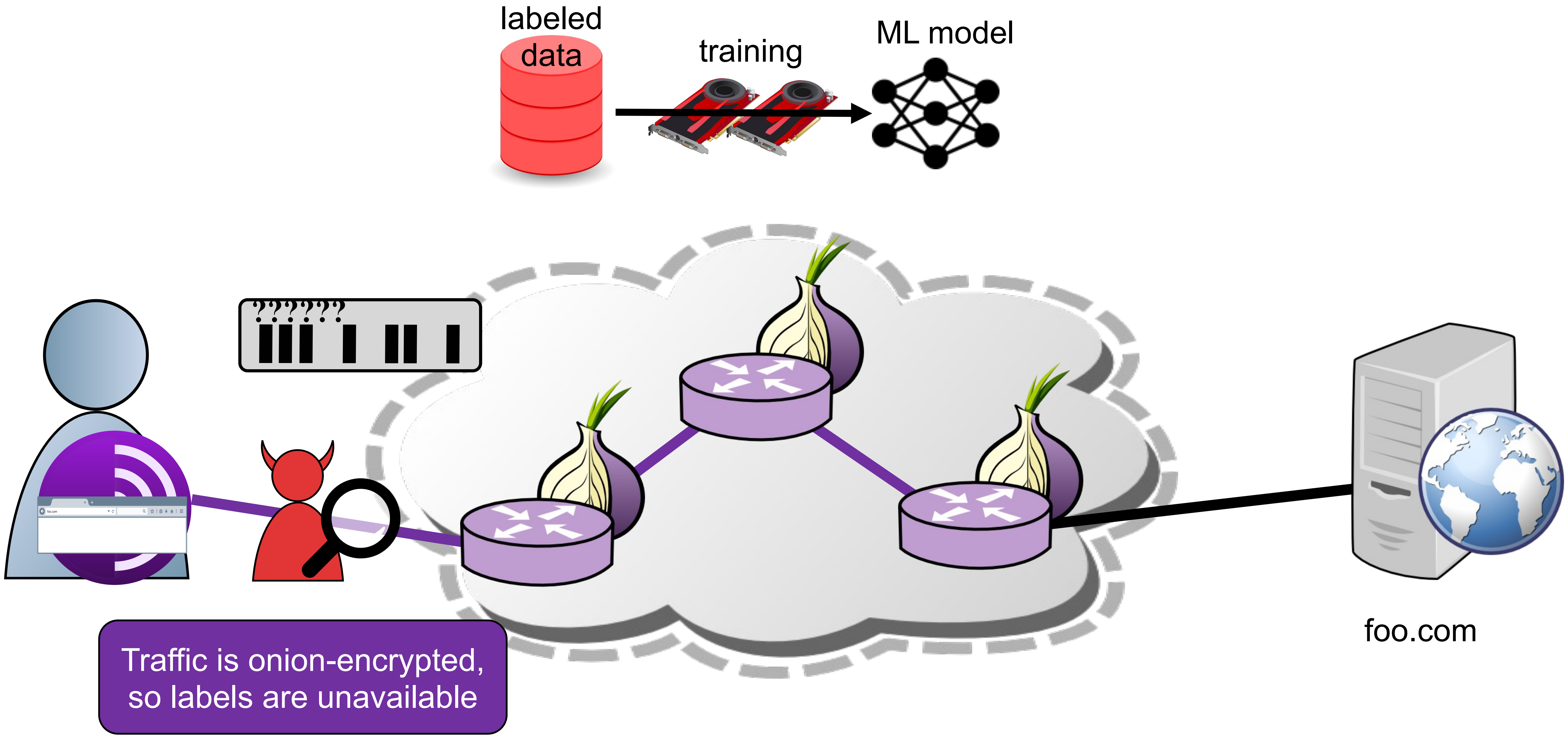


Website Fingerprinting (WF) Threat Model



WF Attacks:
- Observe client-side traffic patterns
- Predict website visited by user, breaking Tor's anonymity

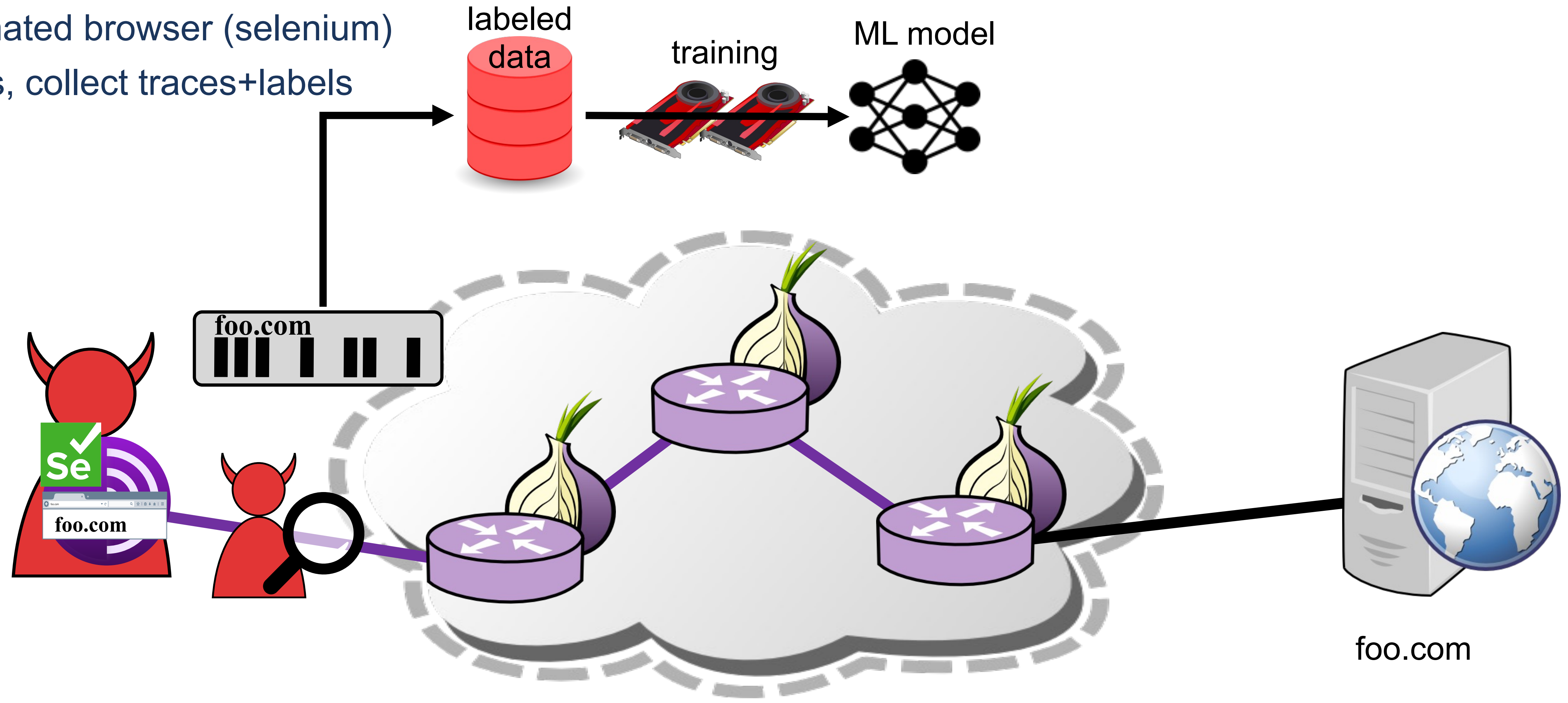
How Might an Adversary Train its ML Models?



How Might an Adversary Train its ML Models?

Traditional method?

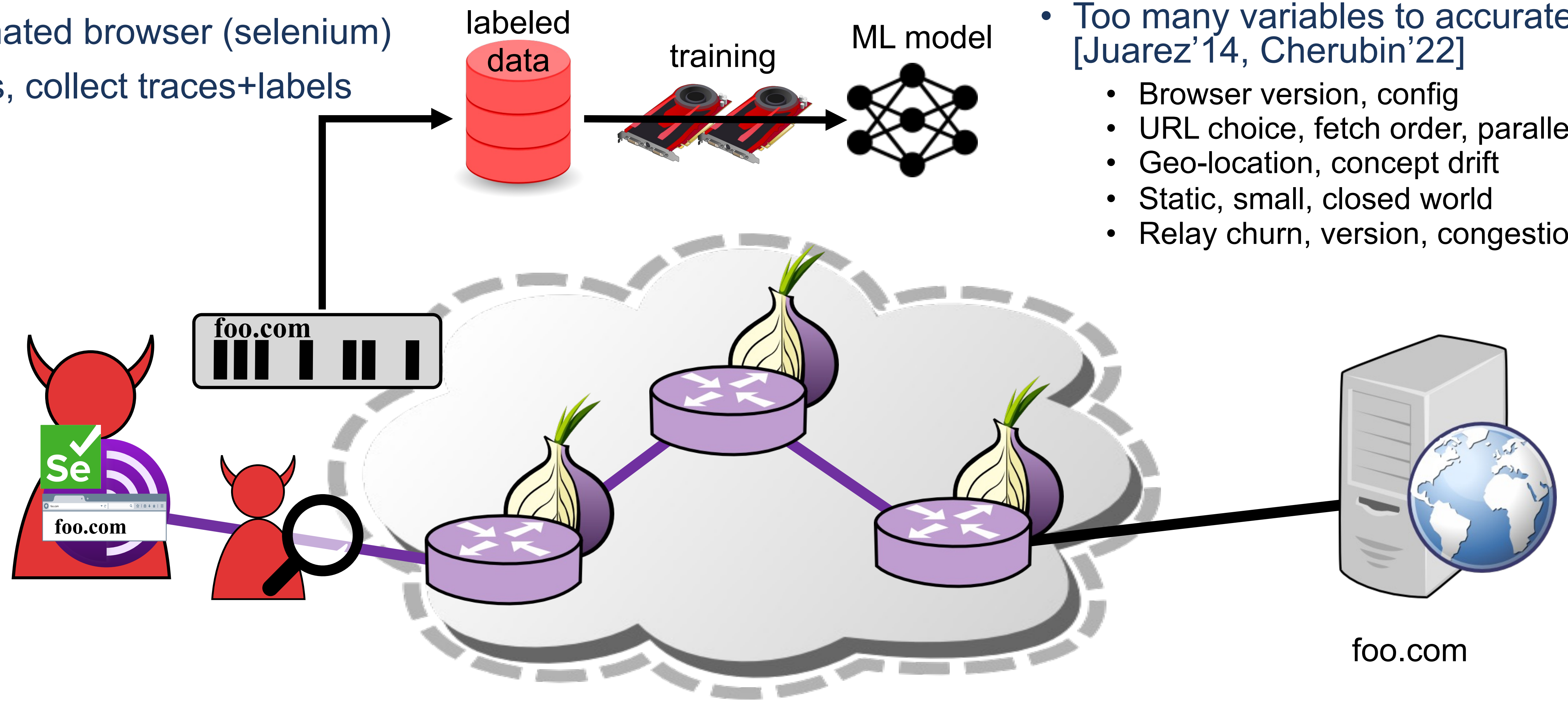
- Use automated browser (selenium)
- Crawl sites, collect traces+labels



How Might an Adversary Train its ML Models?

Traditional method?

- Use automated browser (selenium)
- Crawl sites, collect traces+labels



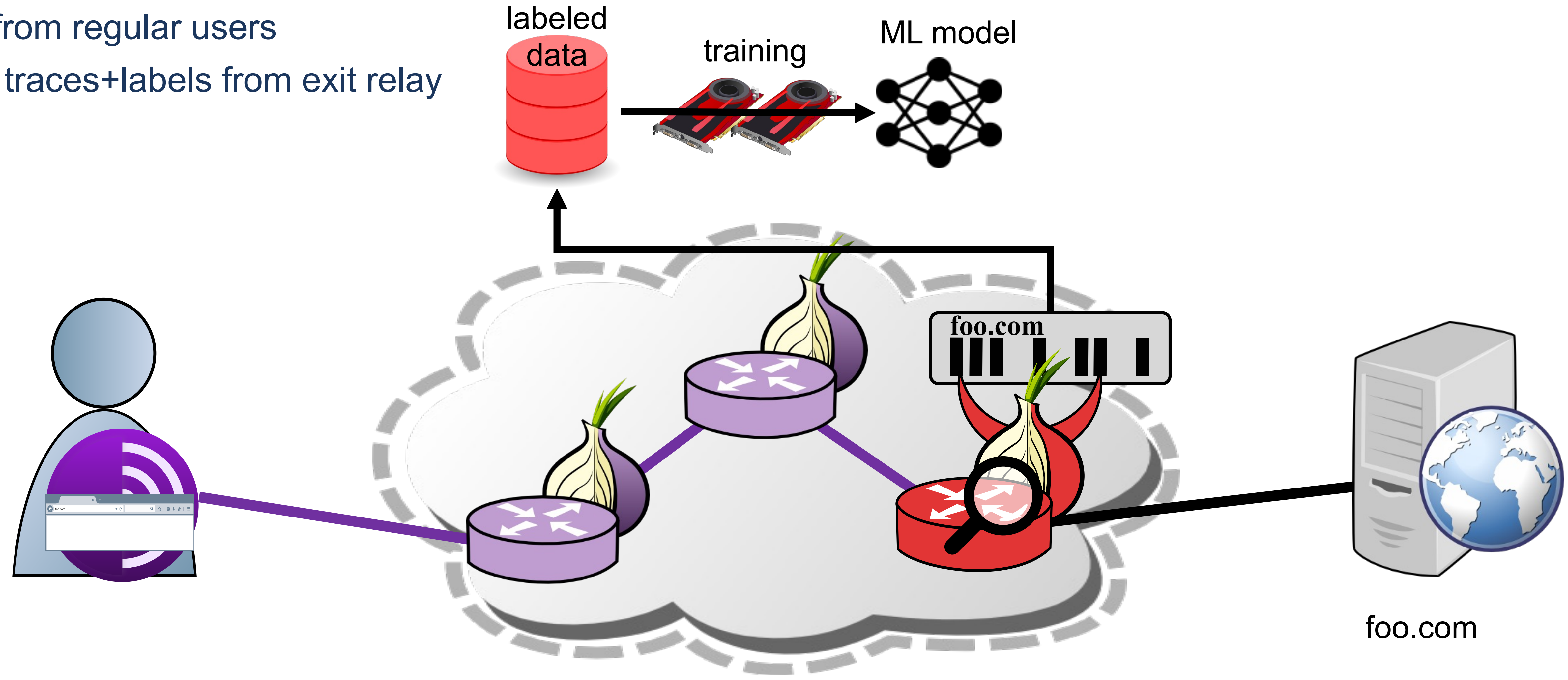
Problems:

- Too many variables to accurately model [Juarez'14, Cherubin'22]
  - Browser version, config
  - URL choice, fetch order, parallel tabs
  - Geo-location, concept drift
  - Static, small, closed world
  - Relay churn, version, congestion, etc.

How Might an Adversary Train its ML Models?

Emerging exit method?

- Traffic from regular users
- Collect traces+labels from exit relay



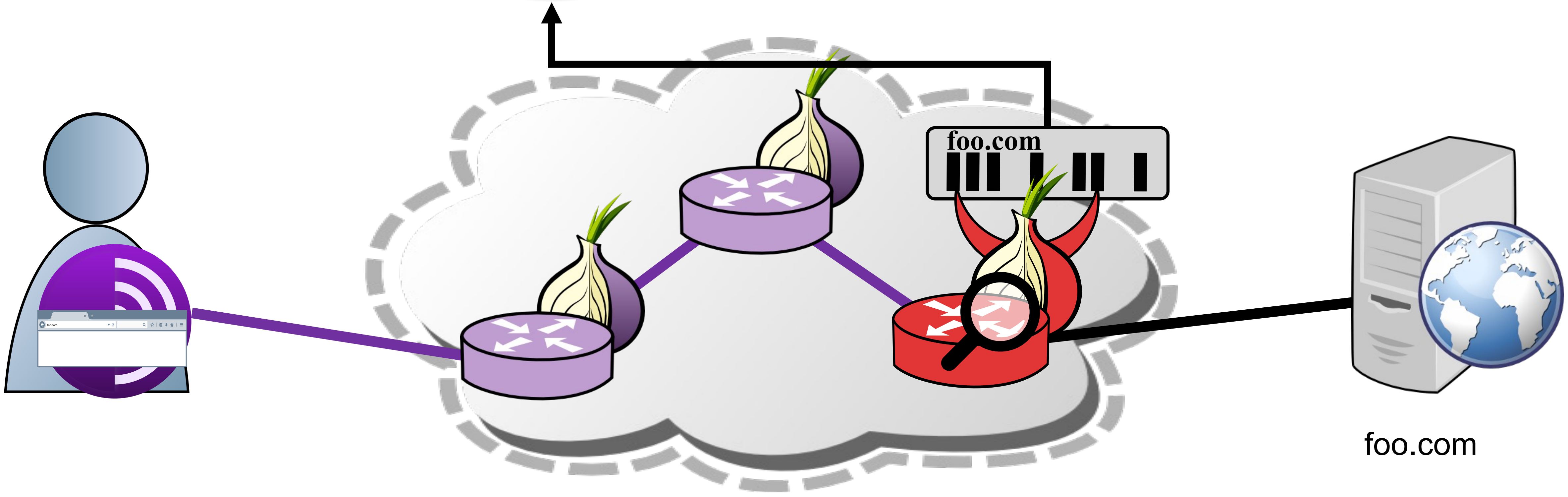
How Might an Adversary Train its ML Models?

Emerging exit method?

- Traffic from regular users
- Collect traces+labels from exit relay

Problems:

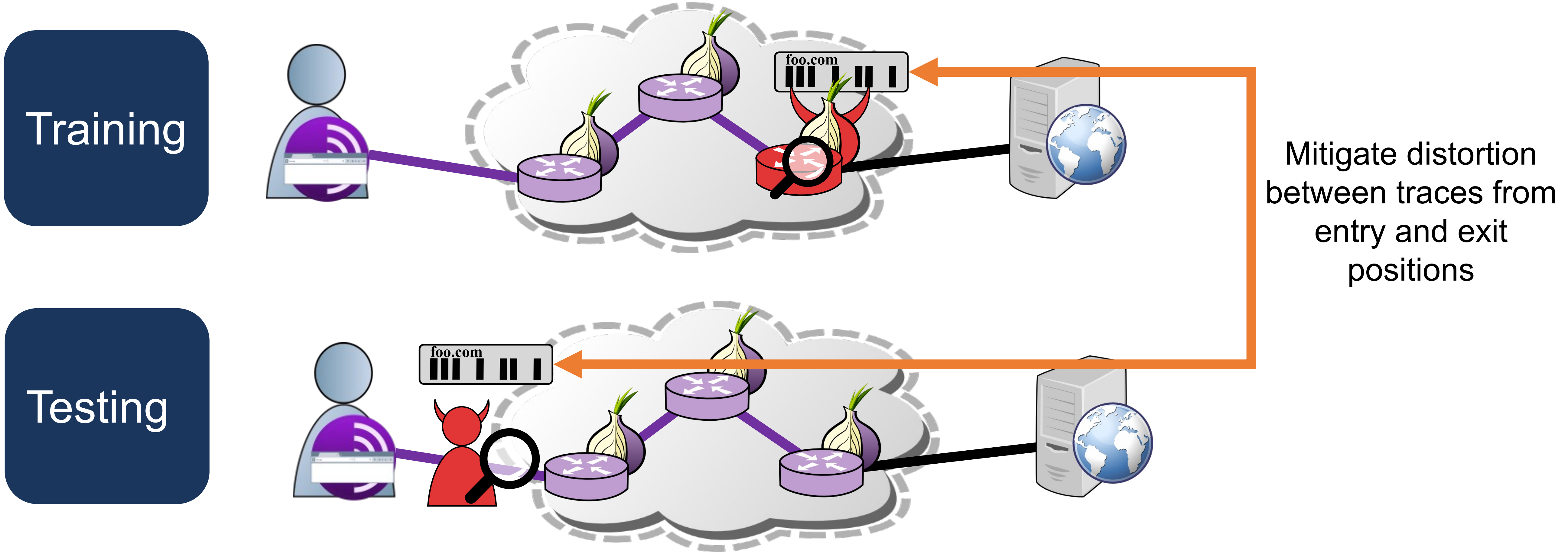
- Training data is collected on *exit*, but testing must be done on *entry*
- Trace “distortion” reduces performance by 5-18% [Cherubin’22]



Research Direction

Research Question:

- How can we mitigate trace distortion so that we can utilize real-world traces to better estimate the threat of WF against Tor?



Outline

1. Trace transduction with Retracer
2. Retracer evaluation
3. Real-world WF evaluation

Cell Trace Transduction

Cell Trace Transduction

- Cell trace:
  - a sequence of *n* (*timestamp*, *direction*) pairs
    - timestamp: when cell was observed, relative to start of connection
    - direction: +1 if forwarded toward server, -1 if toward client

Example cell trace:

[
    (0.1, +1),
    (0.5, -1),
    (0.9, +1),
    (1.3, -1),
    (1.3, -1),
    (1.3, -1),
    …
]

Cell Trace Transduction

Cell Trace Transduction

- Cell trace:
  - a sequence of *n* (*timestamp*, *direction*) pairs
    - timestamp: when cell was observed, relative to start of connection
    - direction: +1 if forwarded toward server, -1 if toward client
- Transducer:
  - a function $T(I, M, p_{in}, p_{out}) \rightarrow [O]_M$
  - transforms an input cell trace *I* in position p_{in} into *M* output cell traces *O* in position p_{out}
  - we want p_{in}=exit, p_{out}=entry

Example cell trace:

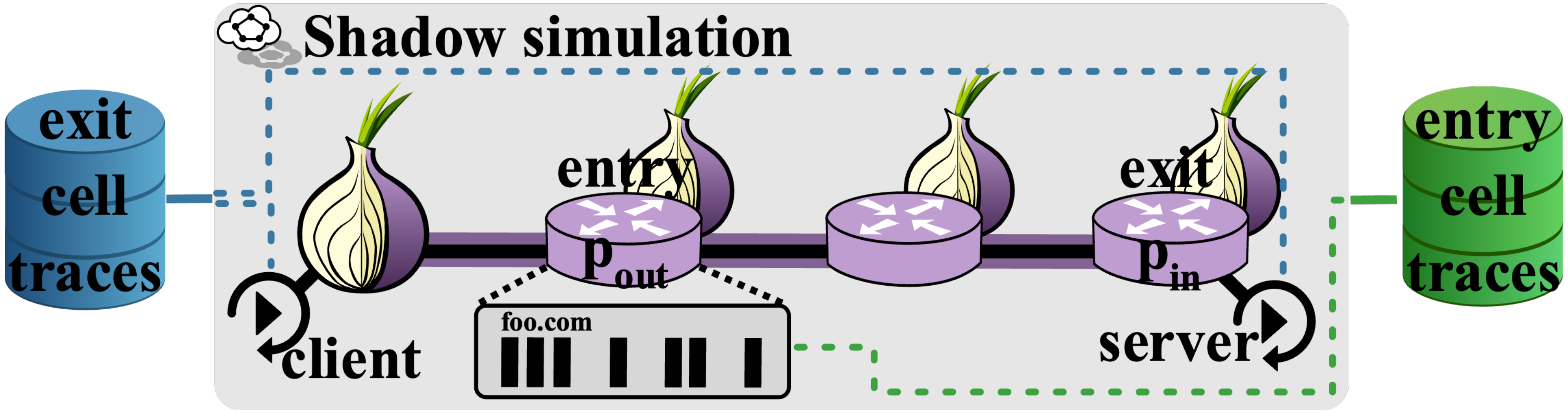
```
[
  (0.1, +1),
  (0.5, -1),
  (0.9, +1),
  (1.3, -1),
  (1.3, -1),
  (1.3, -1),
  …
]
```

Retracer: A Cell Trace Transducer

- Key Insights
  - A cell trace has the metadata needed to reproduce it
  - Network simulation tools (Shadow) model Tor with high fidelity
  - We can replay an *exit* trace in Shadow and extract its *entry* trace

Retracer: A Cell Trace Transducer

- Key Insights
  - A cell trace has the metadata needed to reproduce it
  - Network simulation tools (Shadow) model Tor with high fidelity
  - We can replay an *exit* trace in Shadow and extract its *entry* trace
- Retracer
  - Replays cells traces in large-scale Tor simulations with Shadow
  - Uses cell trace timing and directions as a transcript for replay
  - Adjusts for latency between client and exit during replay

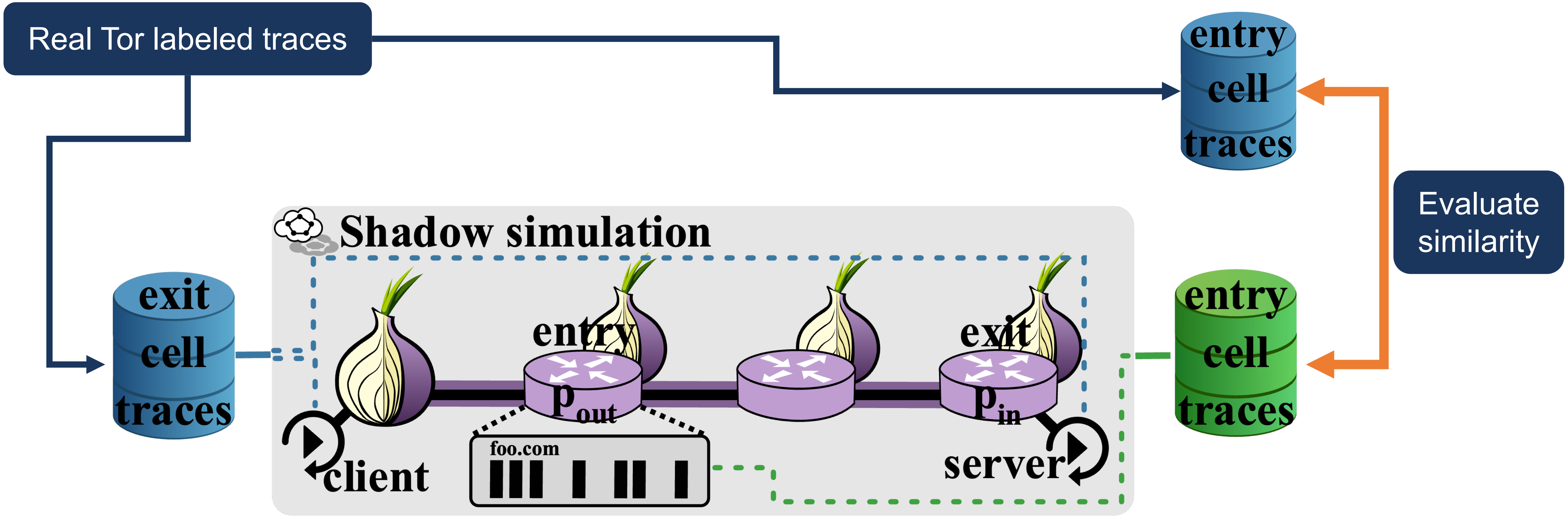


Outline

1. Trace transduction with Retracer
2. Retracer evaluation
3. Real-world WF evaluation

Retracer Evaluation Plan

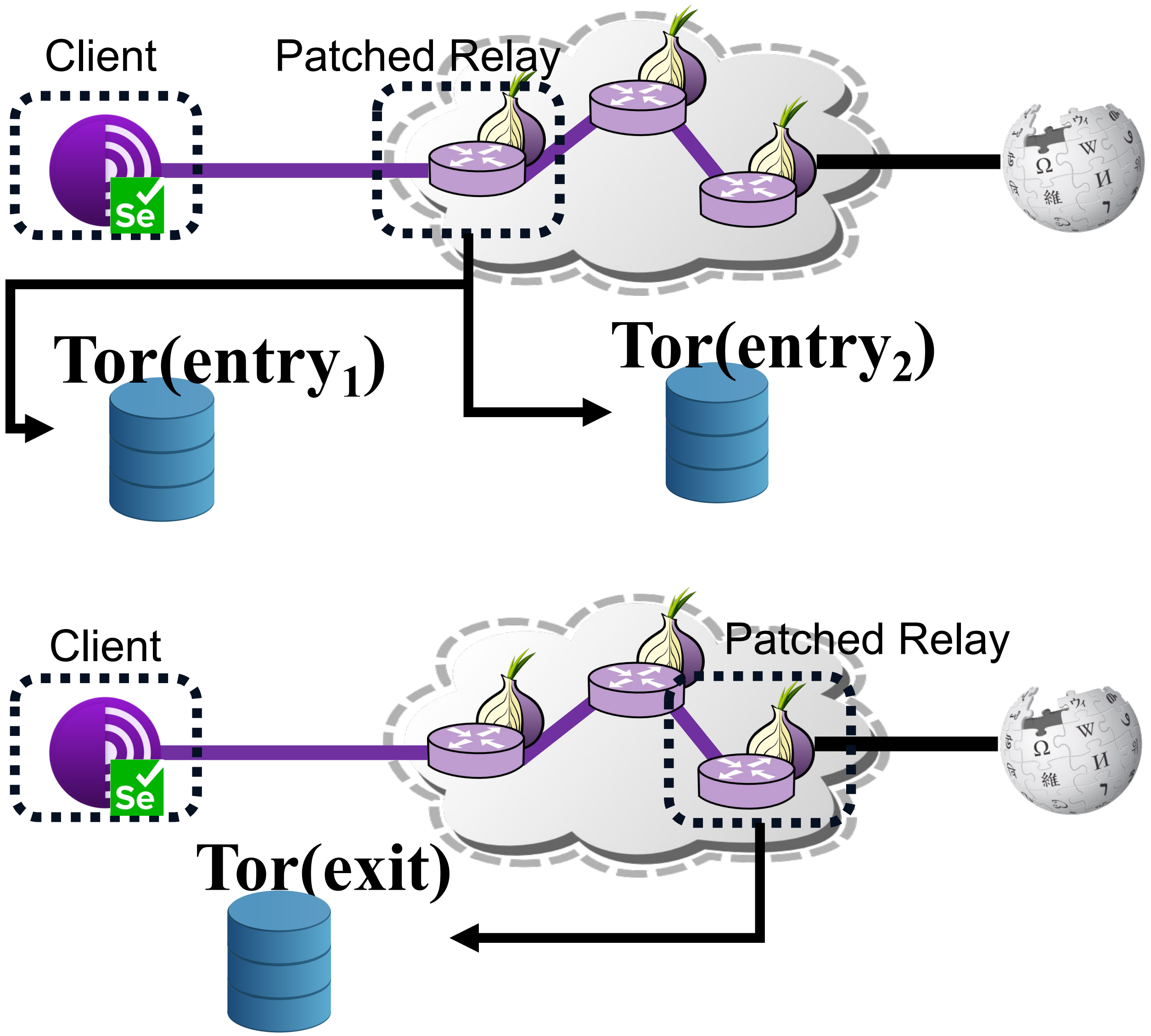
Goal: evaluate how well Retracer transduces *exit* to *entry* traces



Collecting datasets for Retracer evaluation

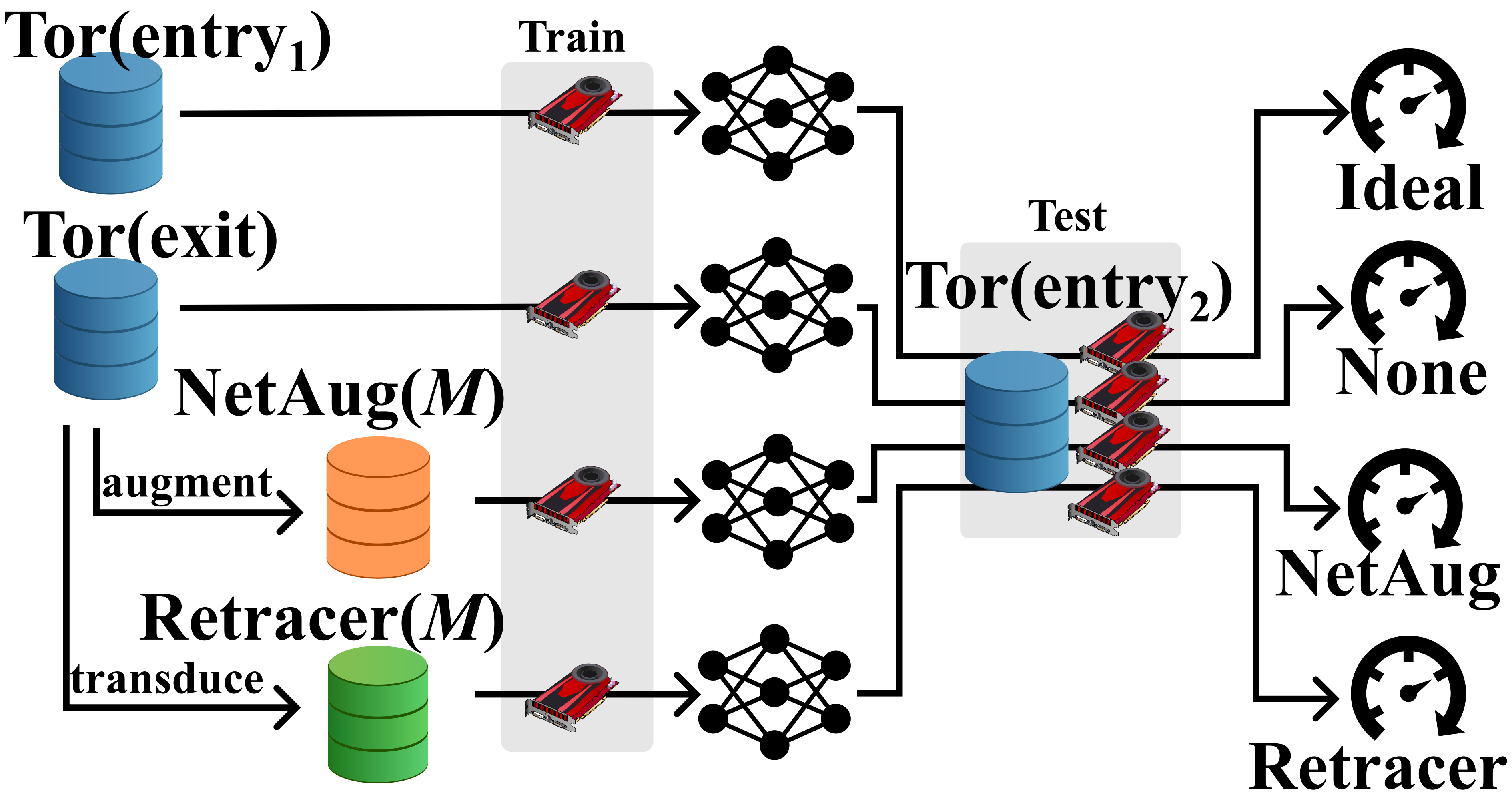
Tor Dataset Collection

- Patch Tor relay to record cell traces (*only* those from *our* client)
- Select some Wikipedia pages
- Fetch each page multiple times through our Tor relay, record traces
- Repeat through Tor exit and entry positions



Retracer Evaluation Methodology

We measure Retracer's efficacy using a downstream WF classification task



Retracer Evaluation Results

Table 2: Classifier Accuracy in a Multiclass Closed World Classification Experiment when Tested on Tor($entry_2$)

Method	Training set	DF	Tik-Tok
Ideal	Tor($entry_1$)	89%	87%
Retracer	Retracer(19)	86% (↓ 3 pp)	85% (↓2 pp)
NetAug	NetAug(19)	70% (↓19 pp)	⊥
None	Tor(exit)	76% (↓13 pp)	79% (↓8 pp)
Classifier Properties →		Time-Oblivious	Time-Aware

⊥: Timing information required by classifier but unavailable in data.

Retracer Evaluation Results

Table 2: Classifier Accuracy in a Multiclass Closed World Classification Experiment when Tested on Tor($entry_2$)

Method	**Training set**	**DF**	**Tik-Tok**
Ideal	Tor($entry_1$)	89%	87%
Retracer	Retracer(19)	86% (↓ 3 pp)	85% (↓2 pp)
NetAug	NetAug(19)	70% (↓19 pp)	⊥
None	Tor(exit)	76% (↓13 pp)	79% (↓8 pp)
Classifier Properties →		Time-Oblivious	Time-Aware

⊥: Timing information required by classifier but unavailable in data.

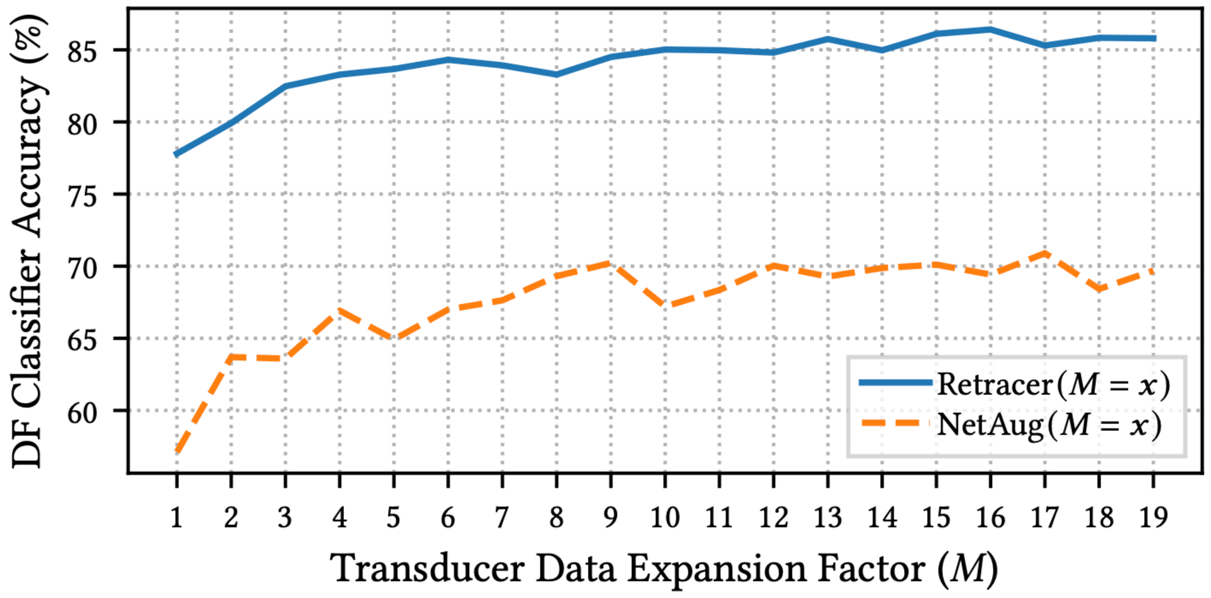


Figure 4: DF classifier accuracy in a multiclass closed-world experiment when training on datasets transduced with an increasing data expansion factor M **and tested on** Tor($entry_2$).

Outline

1. Trace transduction with Retracer

2. Retracer evaluation

3. Real-world WF evaluation

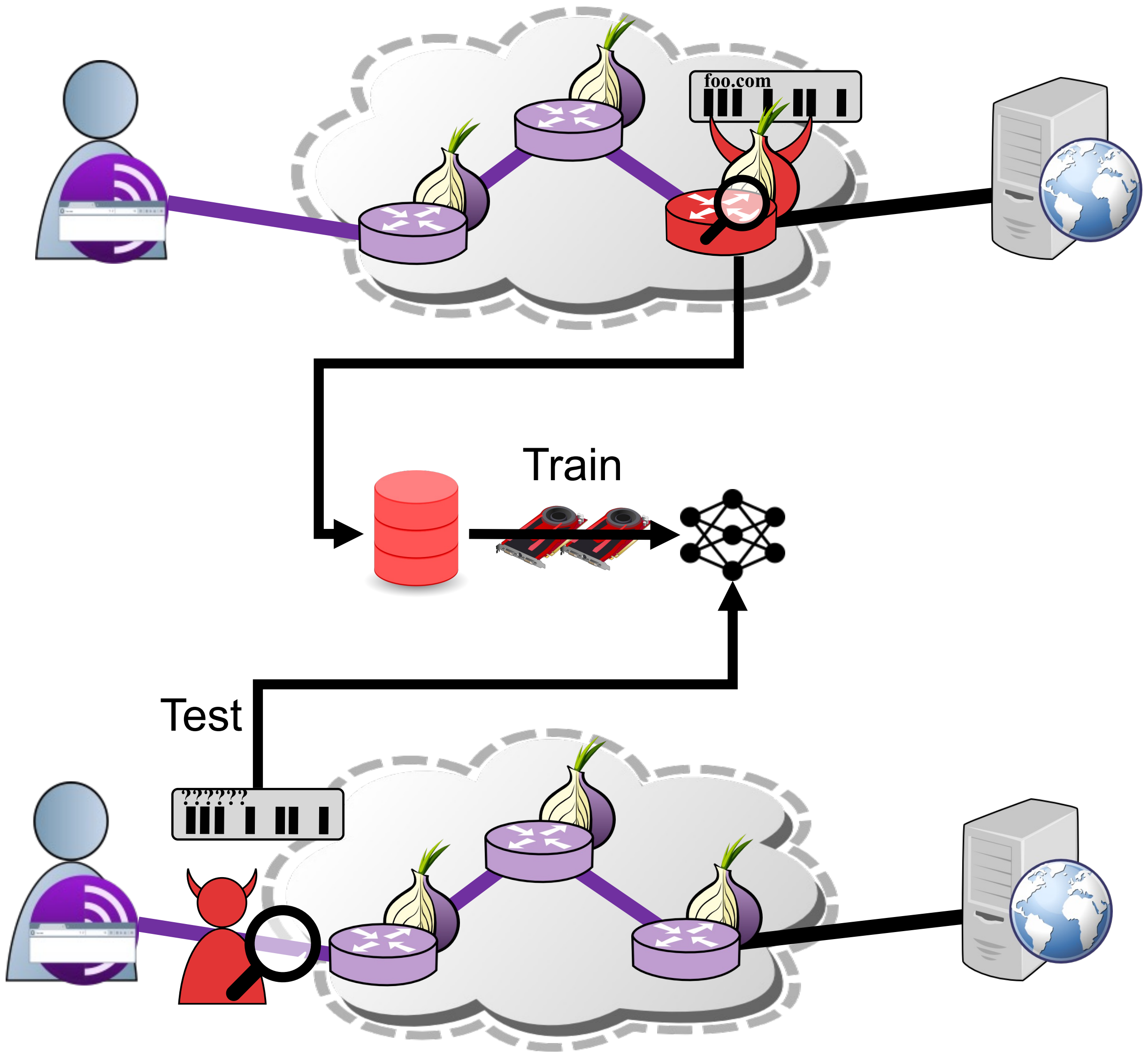
Real-World Evaluation Goals

We consider an adversary that uses real-world traces

- Real: traces from normal Tor users
- Testing ***must*** be against real traces
- Training on real traces is thus superior

We want to estimate WF performance as realistically as possible

- Considering multiple training strategies
- We need a source of real-world data!



U.S. NAVAL RESEARCH LABORATORY

Methodology Considering Genuine Tor Traces

GTT23:

- Contains >13M traces from *real* users
- Collected over 13 weeks on Tor *exits*

GTT23 is available online:
Paper: https://doi.org/10.48550/arXiv.2404.07892
Dataset: https://doi.org/10.5281/zenodo.10620519

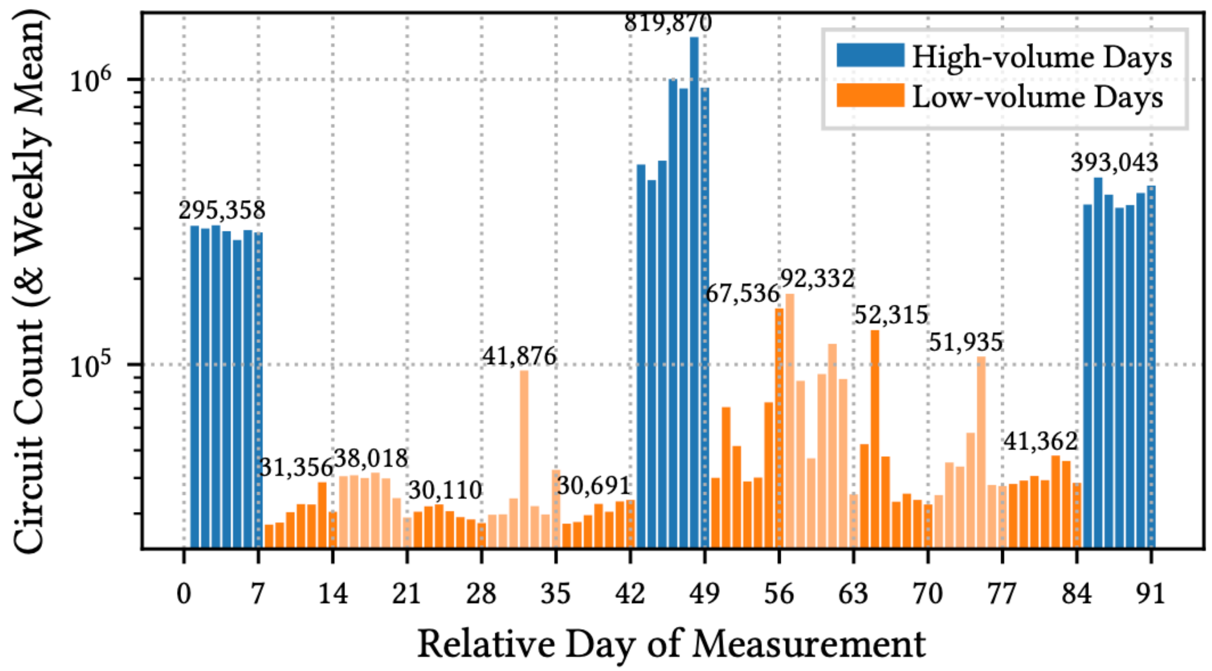


Figure 1: The daily total (bars) and weekly mean (text) number of circuits during our 13 week measurement.

U.S. NAVAL RESEARCH LABORATORY

Methodology Considering Genuine Tor Traces

GTT23:

- Contains >13M traces from *real* users
- Collected over 13 weeks on Tor *exits*

Training:

- Use Deep Fingerprinting (DF) model
- Week 1 traces with ≥ 1000 cells
- 1 model for each of the ~400 most popular websites

Testing

- Traces from weeks >1
- Open world: some sites not trained on

GTT23 is available online:
Paper: https://doi.org/10.48550/arXiv.2404.07892
Dataset: https://doi.org/10.5281/zenodo.10620519

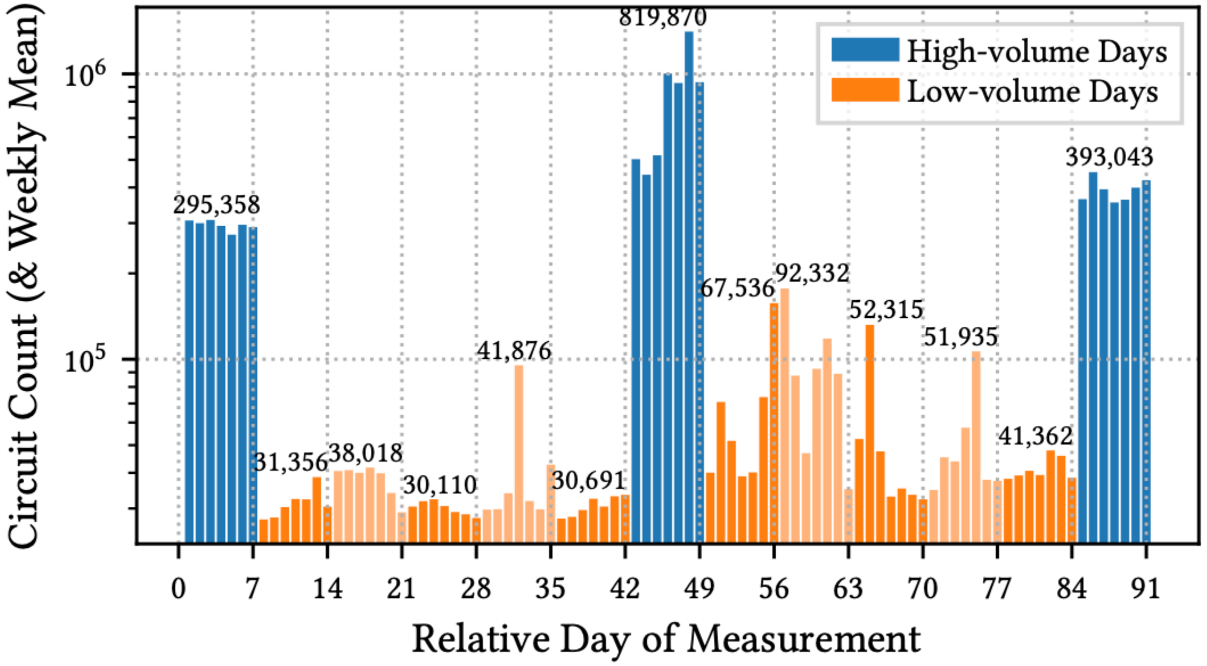
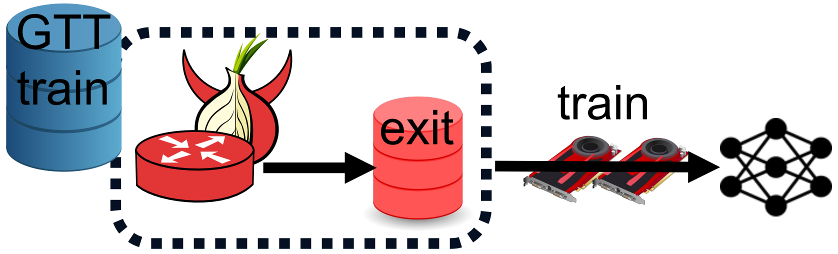


Figure 1: The daily total (bars) and weekly mean (text) number of circuits during our 13 week measurement.

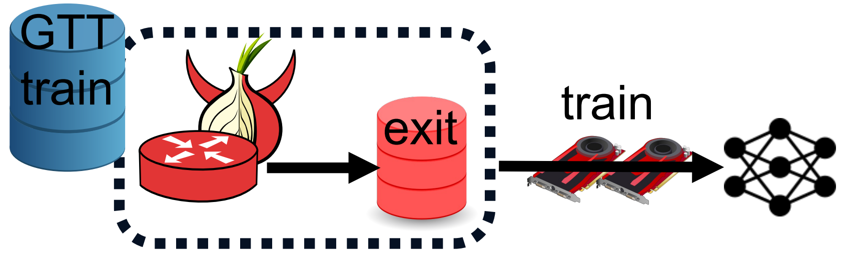
WF Performance when Testing on Entry Traces

OnlineWF Train: (Cherubin'22)

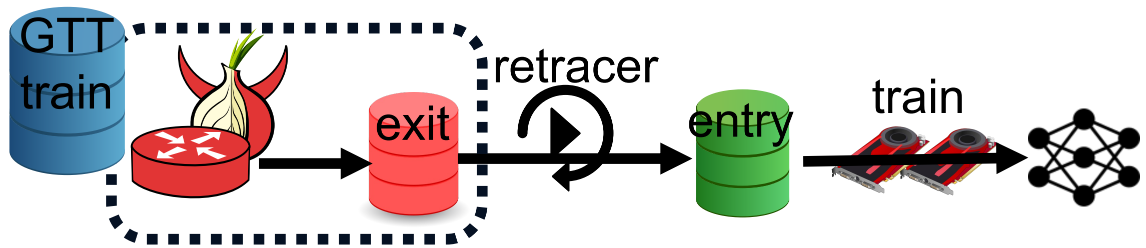


WF Performance when Testing on Entry Traces

OnlineWF Train: (Cherubin'22)

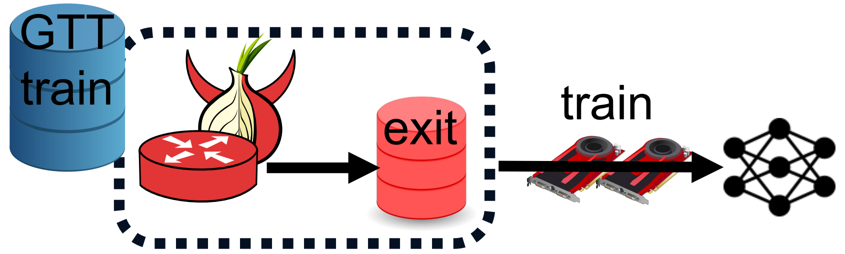


Retracer Train:

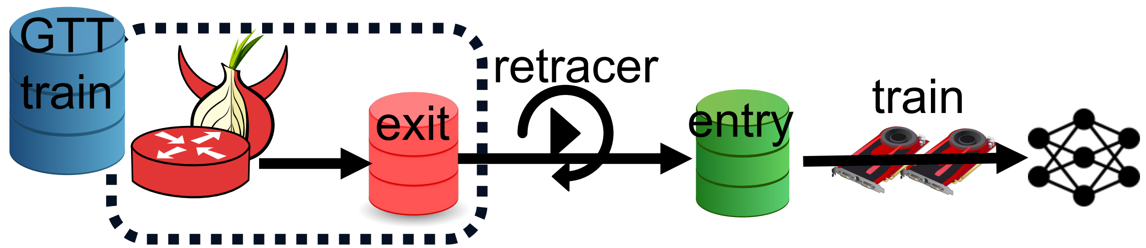


WF Performance when Testing on Entry Traces

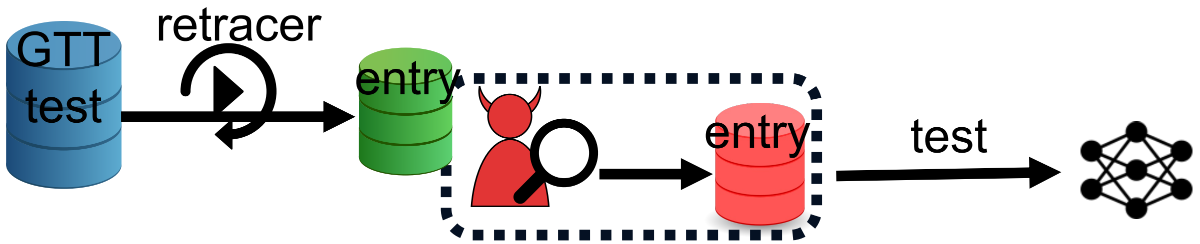
OnlineWF Train: (Cherubin'22)



Retracer Train:

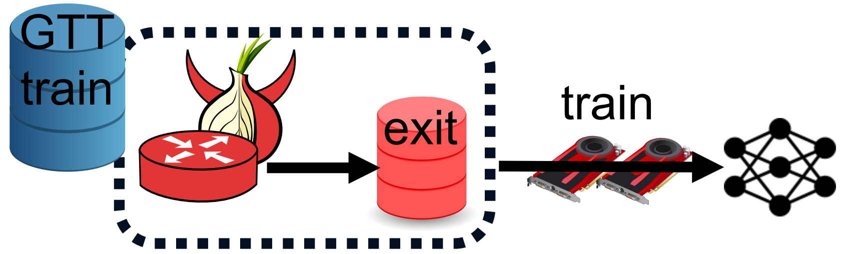


Both Test:

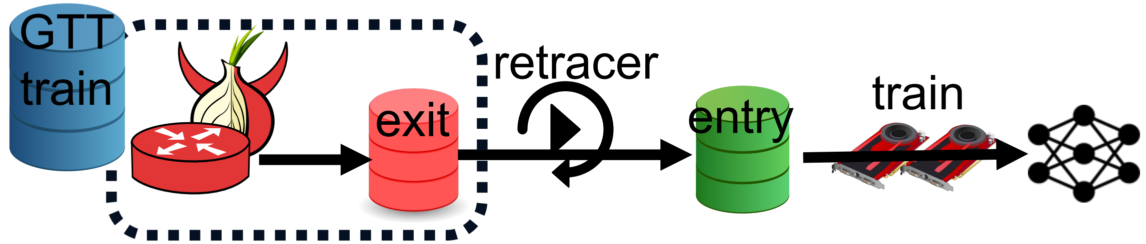


WF Performance when Testing on Entry Traces

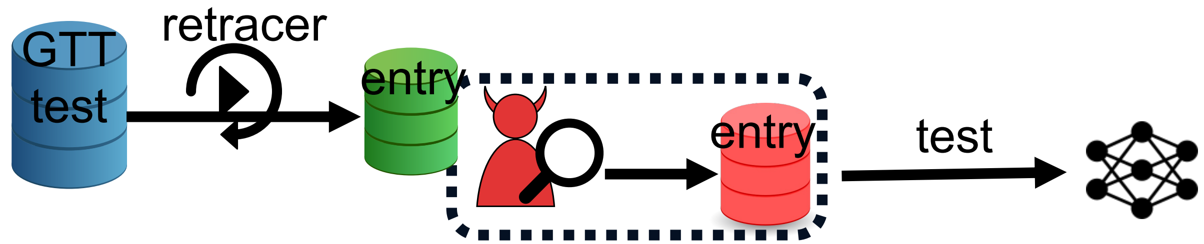
OnlineWF Train: (Cherubin'22)



Retracer Train:



Both Test:



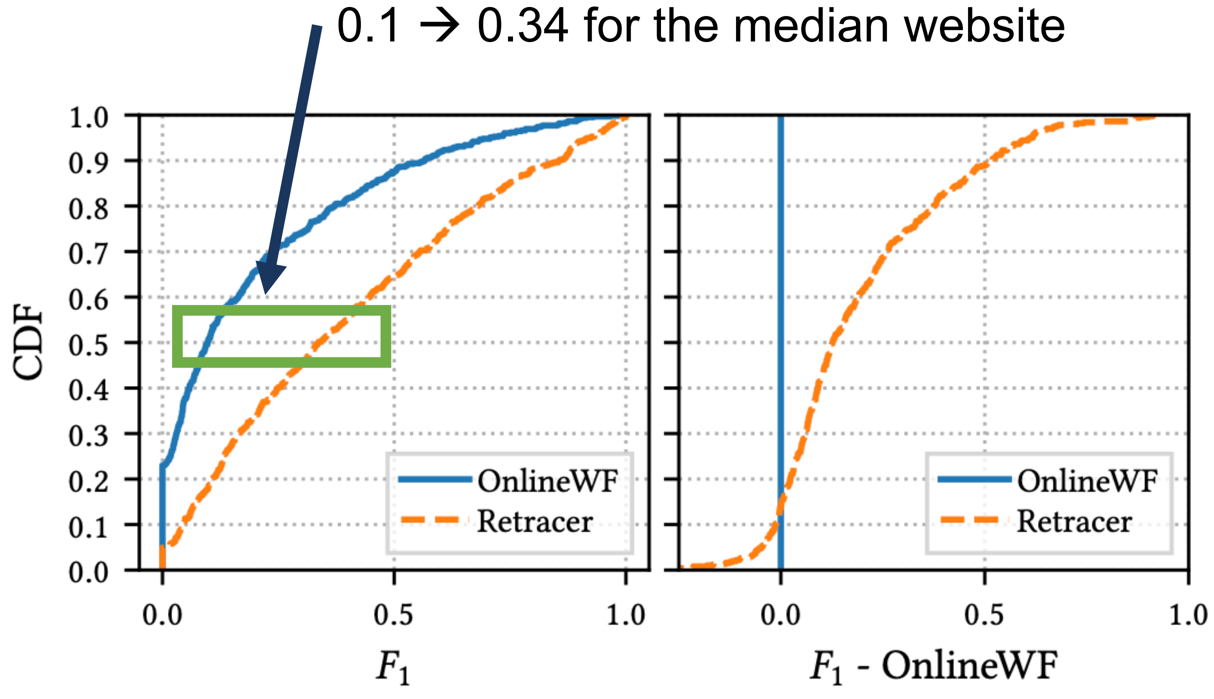


Figure 8: Classifier performance when training on exit traces as in OnlineWF [8] and training on entry traces transduced from the exit traces by Retracer.

Synthetic Datasets Overestimate WF Performance

Retracer: trained & tested as before

- Uses Retracer to transduce the GTT23 train and test sets

Synthetic datasets → previous work

- BigEnough: ~100,000 traces
- GoodEnough: ~10,000 traces
- Multiple pages per site
- Use analogous per-site training/testing methodology

Synthetic Datasets Overestimate WF Performance

Retracer: trained & tested as before

- Uses Retracer to transduce the GTT23 train and test sets

Synthetic datasets → previous work

- BigEnough: ~100,000 traces
- GoodEnough: ~10,000 traces
- Multiple pages per site
- Use analogous per-site training/testing methodology

WF performs better with synthetic traces

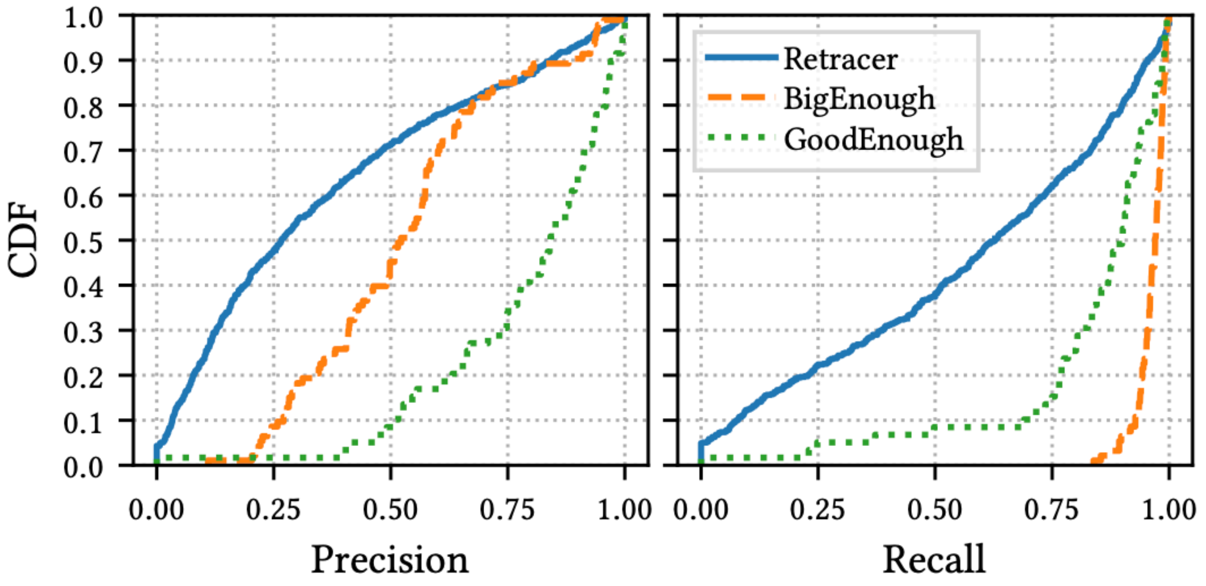
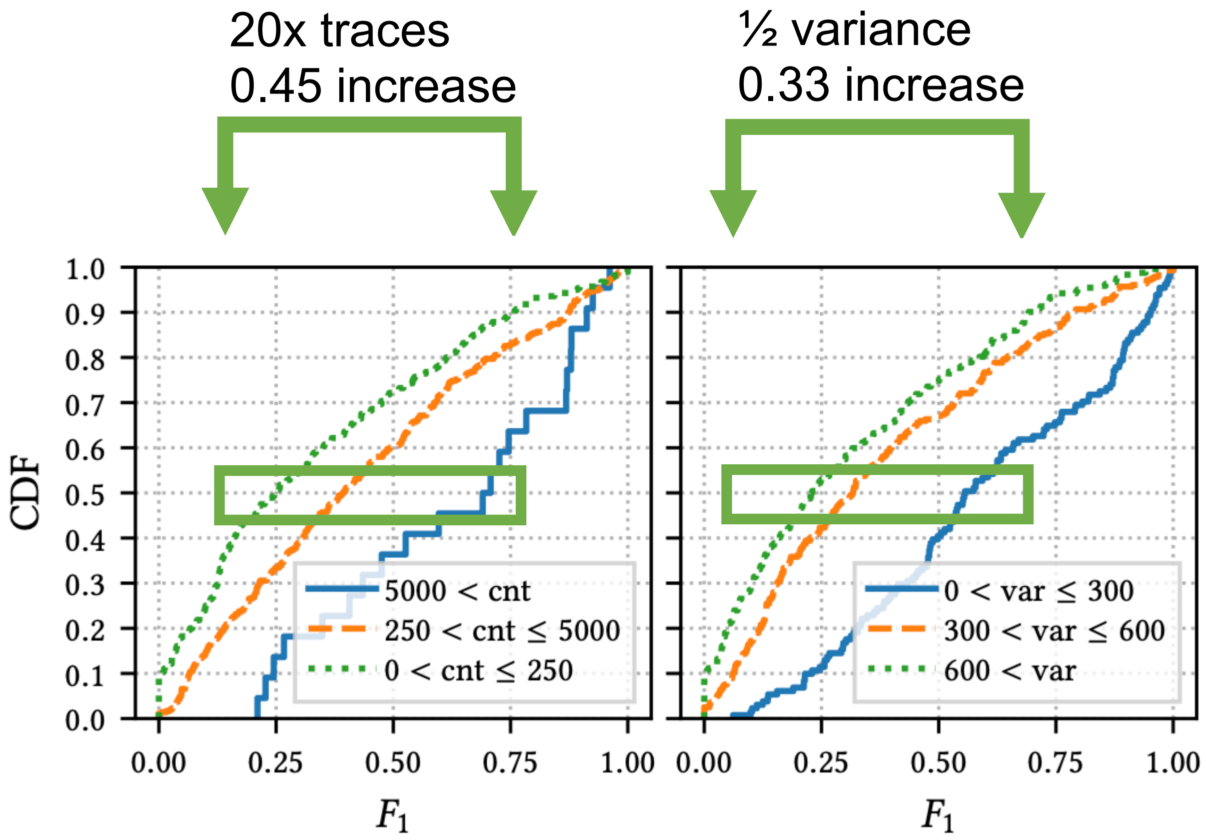


Figure 9: Performance of the classifiers trained and tested with each dataset. "Synthetic" traces lead to better performance than Retracer traces (transduced from GTT23).

What are the important features for performance?

Feature importance analysis – features predicting performance

1. Trace count
   - Median F_1 increased by 0.45 when 20x as many traces were available
2. Variance of trace lengths
   - Median F_1 increased by 0.33 when half as much variance is observed



U.S. NAVAL RESEARCH LABORATORY

Repositioning Real-World Website Fingerprinting on Tor

Read the paper!

Contributions

- Retracer for transducing cell traces across positions
- Retracer evaluation using Tor datasets
- Real-world WF evaluation that tests on entry traces
- Individual website fingerprintability methodology
- Feature importance analysis

Future Work

- Use Retracer to evaluate WF in new scenarios
  - Traffic splitting with Conflux
  - Apply WF defenses on top of genuine data

Contact:
robert.g.jansen7.civ@us.navy.mil
robgjansen.com