U.S. NAVAL RESEARCH LABORATORY

Data-Explainable Website Fingerprinting with Network Simulation

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

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

The 23rd Privacy Enhancing Technologies Symposium
Lausanne, Switzerland
July 11th, 2023

U.S. NAVAL RESEARCH LABORATORY

sexy version:

How to Accelerate Website Fingerprinting Research!

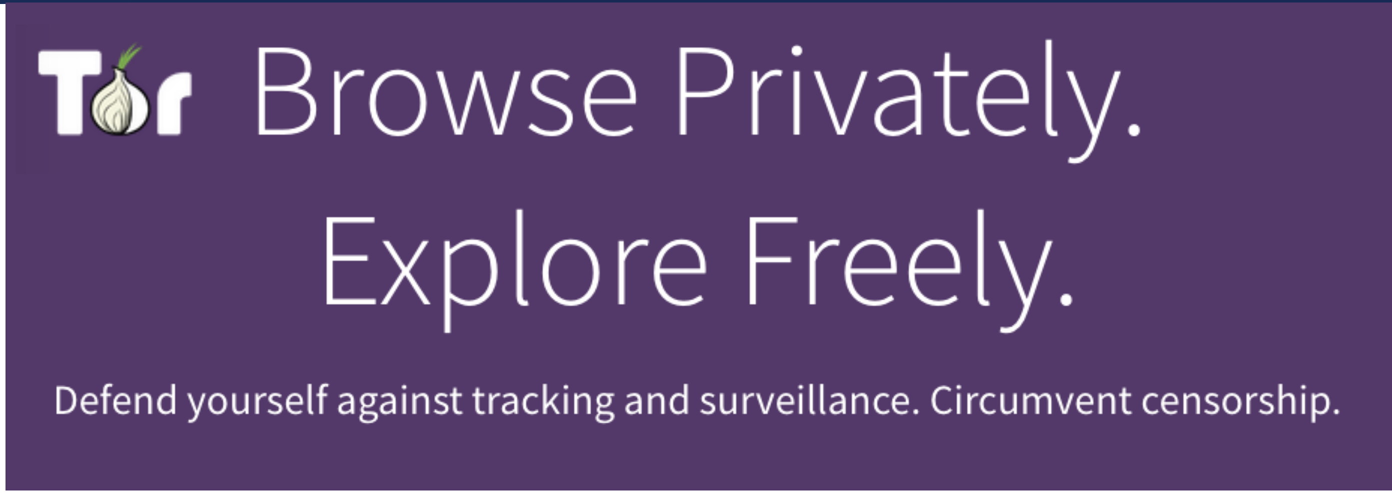
Hint: stop crawling Tor to gather datasets, use network simulation instead

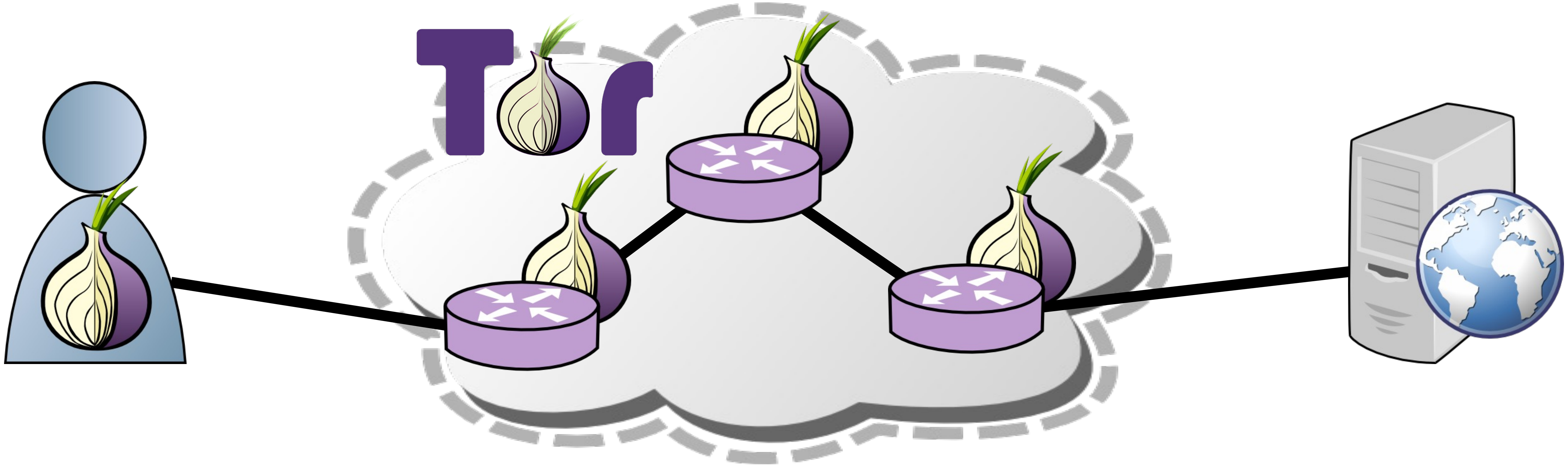
- perfectly privacy preserving, no risk/load on Tor
- unlimited source of accurately labeled data
- higher data diversity
- controlled network → explainable data
- simulation-assisted WF outperforms standard methods

U.S. NAVAL RESEARCH LABORATORY

Anonymous Communication with Tor

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

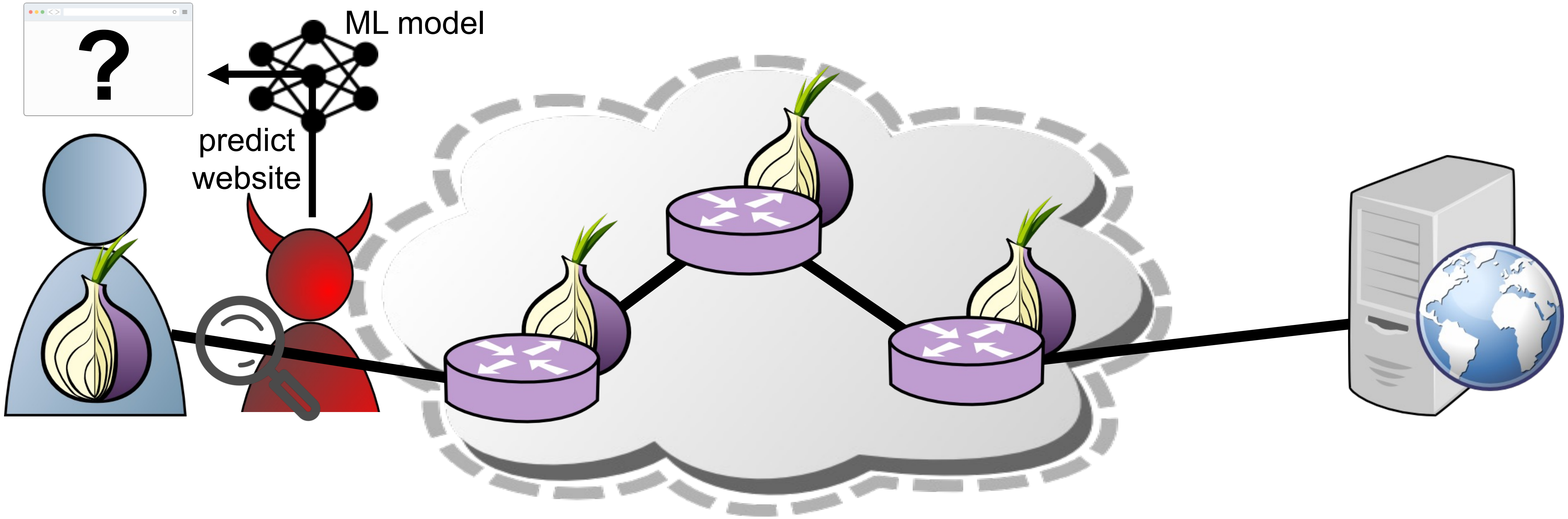




Website Fingerprinting (WF) Threat Model

<u>WF Attacks</u>:

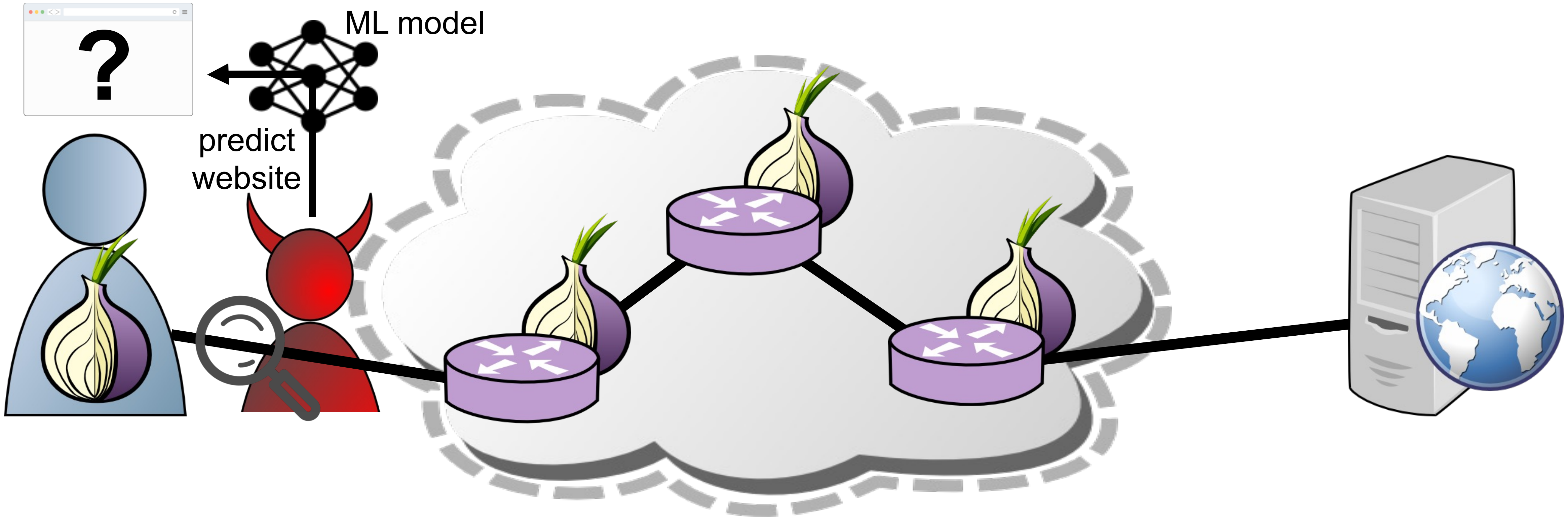
- Predict website visited by user
- Break Tor's anonymity



Website Fingerprinting (WF) Threat Model

WF Attacks:
- Predict website visited by user
- Break Tor's anonymity

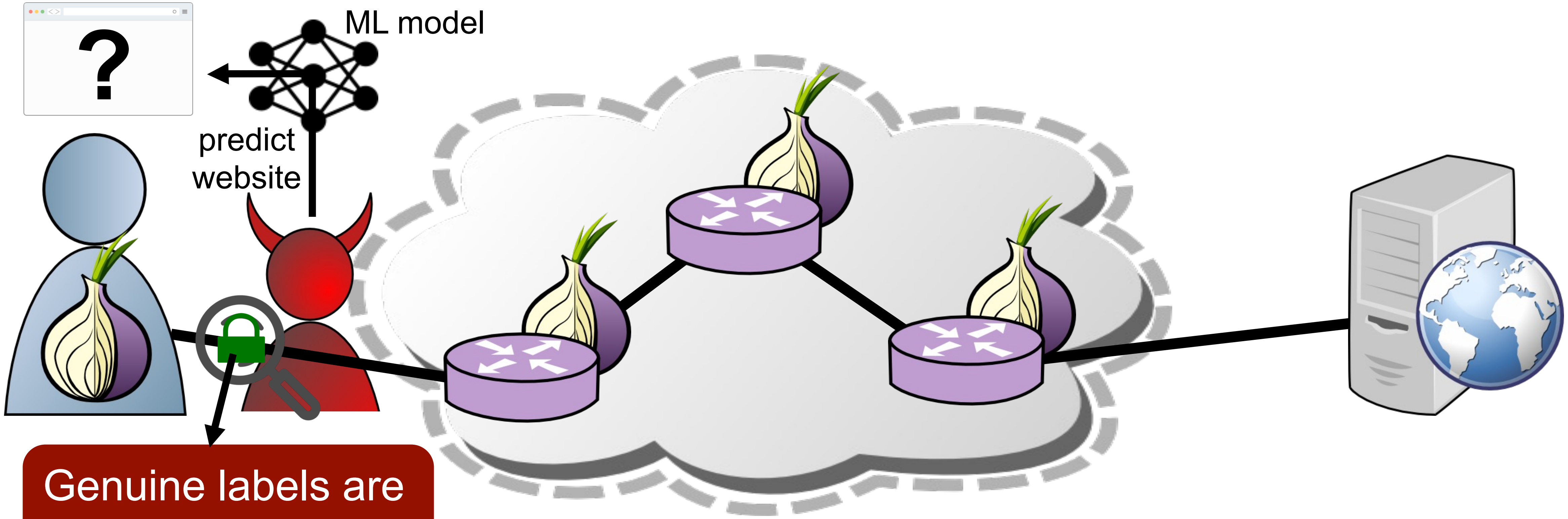
Requirements:
- Observe entry-side packet traces
- Labeled data to train ML models



Website Fingerprinting (WF) Threat Model

WF Attacks:
- Predict website visited by user
- Break Tor's anonymity

Requirements:
- Observe entry-side packet traces
- Labeled data to train ML models

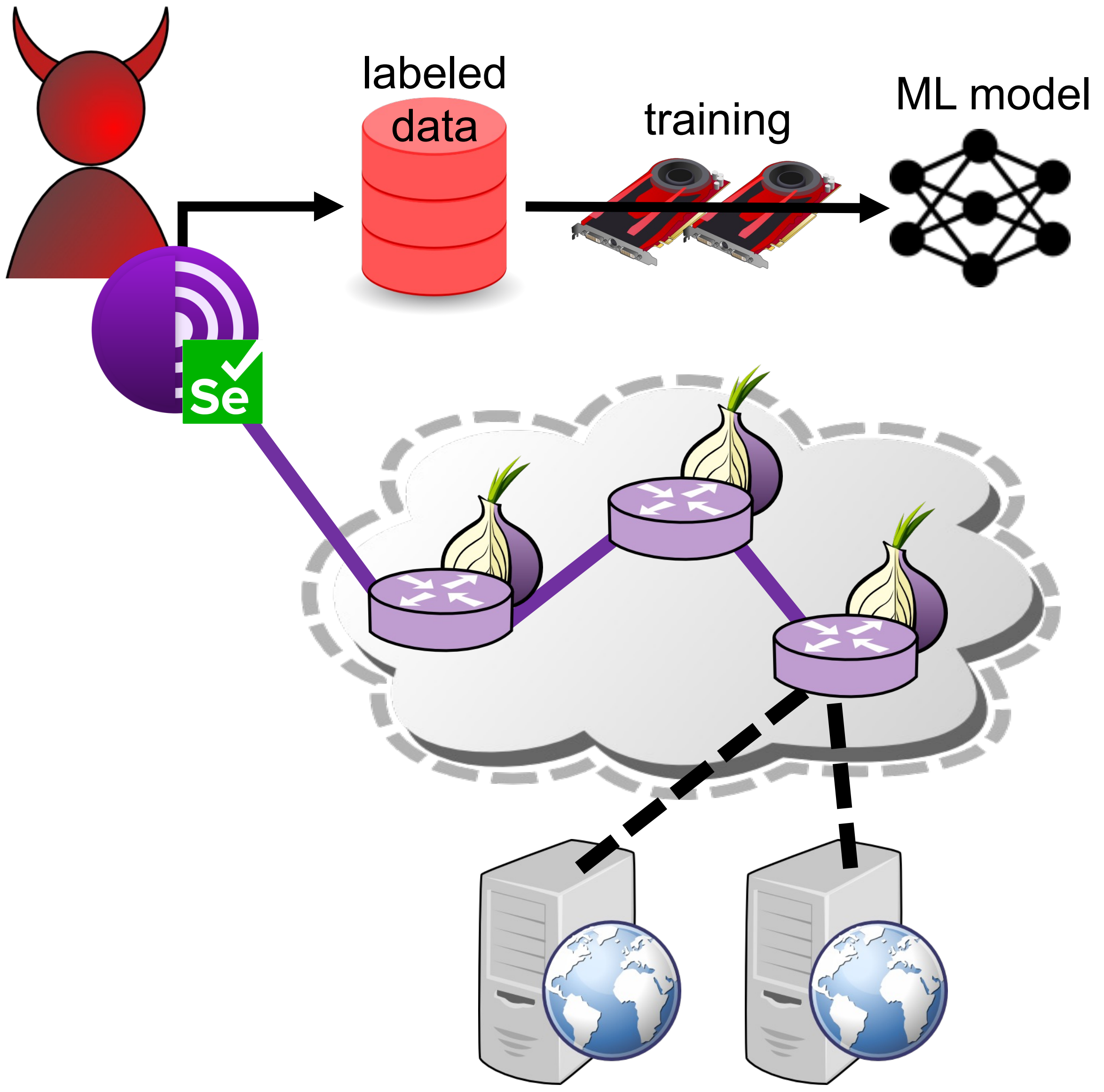


Genuine labels are encrypted by Tor

How Might an Adversary Train its ML Models?

Traditional method:
(used almost exclusively in WF research)

- Use automated browser (selenium)
- Crawl sites, collect traces+labels
- Train ML models offline
- Repeat continuously to stay fresh



How Might an Adversary Train its ML Models?

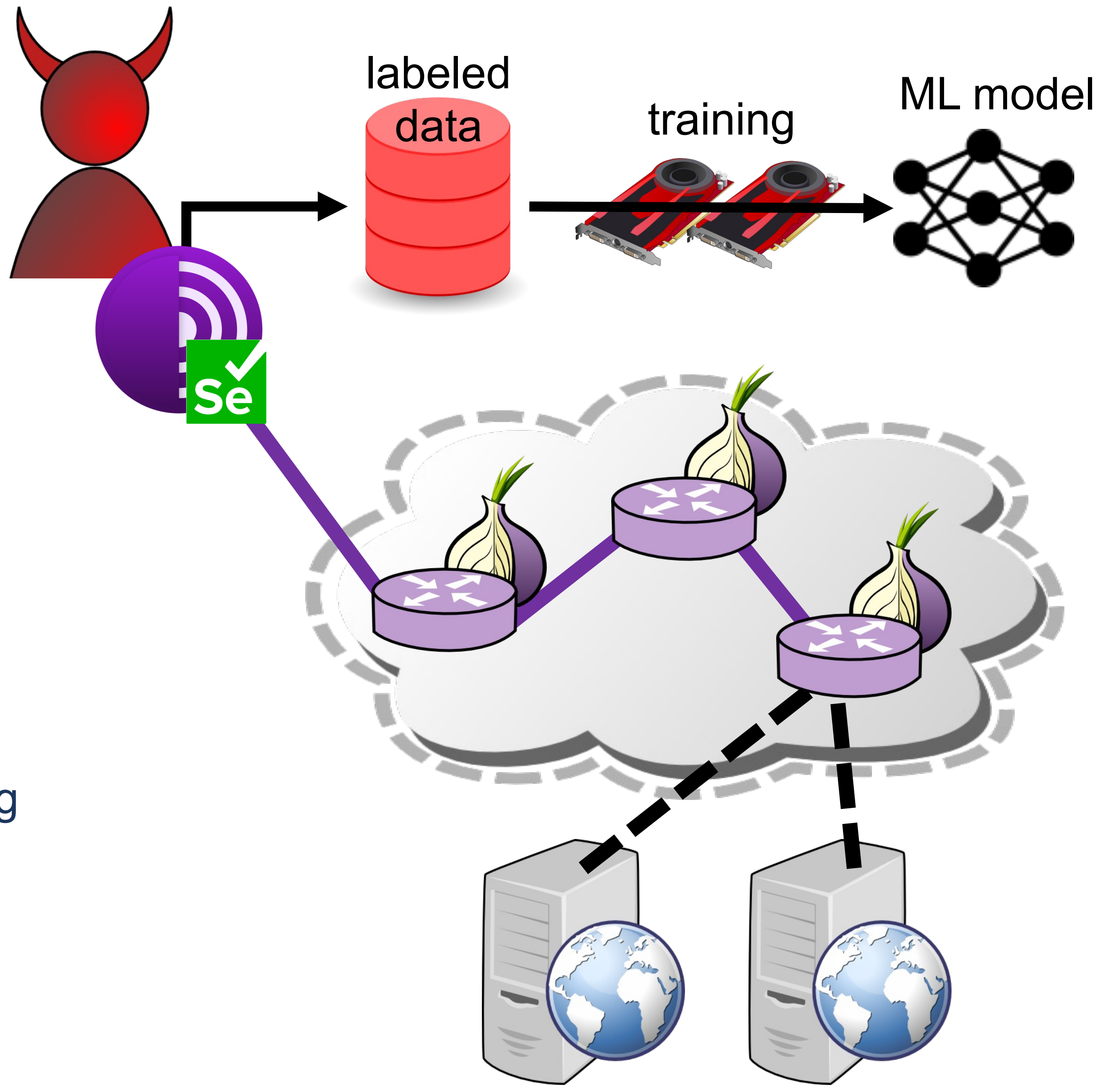
Traditional method:
(used almost exclusively in WF research)

- Use automated browser (selenium)
- Crawl sites, collect traces+labels
- Train ML models offline
- Repeat continuously to stay fresh

Problem: low-quality datasets!
(many variables affect data quality)

- Browser version, config
- URL choice, fetch order
- Use of parallel tabs
- Geo-location
- Data staleness
- Static, small, closed world

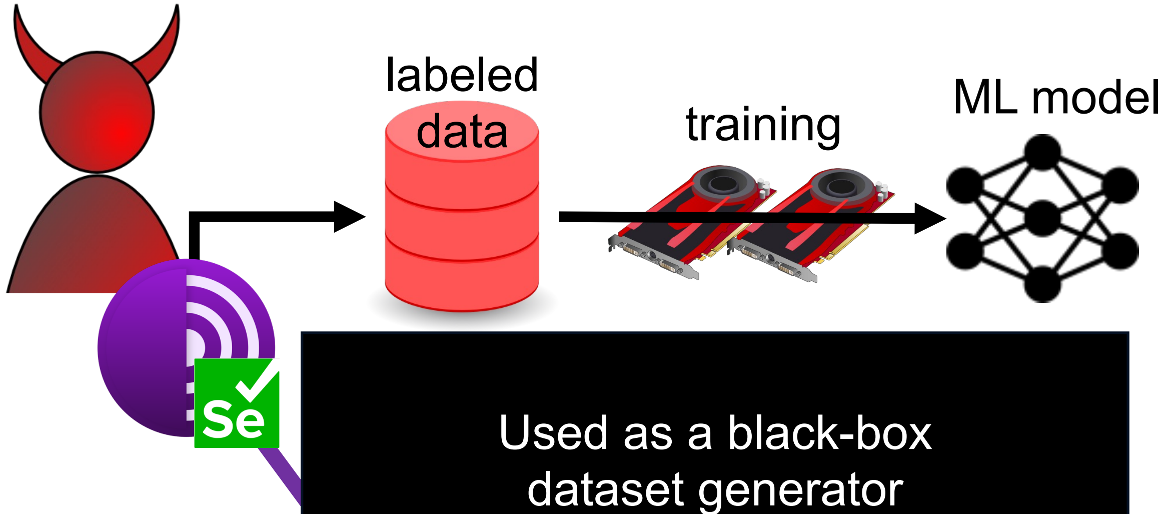
- Relay churn, version, config
- Relay congestion
  - Network usage fluctuations
  - Low bandwidth relays
  - DoS attacks
  - exit port exhaustion
  - censorship events



How Might an Adversary Train its ML Models?

Traditional method:
(used almost exclusively in WF research)

- Use automated browser (selenium)
- Crawl sites, collect traces+labels
- Train ML models offline
- Repeat continuously to stay fresh



Used as a black-box
dataset generator

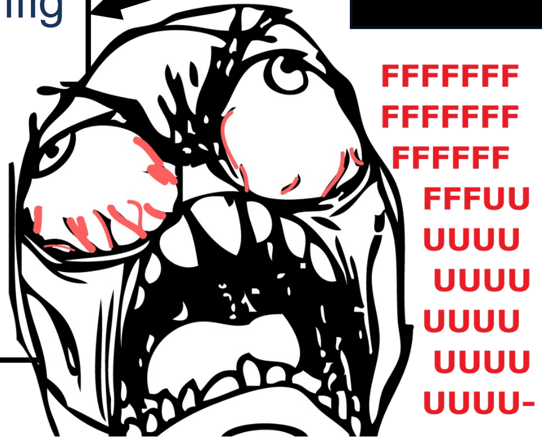
BUT

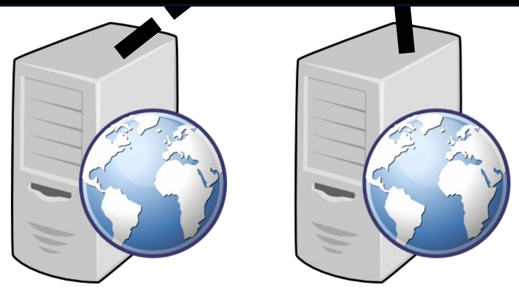
we have little control over Tor
and don't really understand the data!

Problem: low-quality datasets!
(many variables affect data quality)

- Browser version, config
- URL choice, fetch order
- Use of parallel tabs
- Geo-location
- Data staleness
- Static, small, closed world

- Relay churn, version, config
- Relay congestion
  - Network usage fluctuations
  - Low bandwidth relays
  - DoS attacks
  - exit port exhaustion
  - censorship events

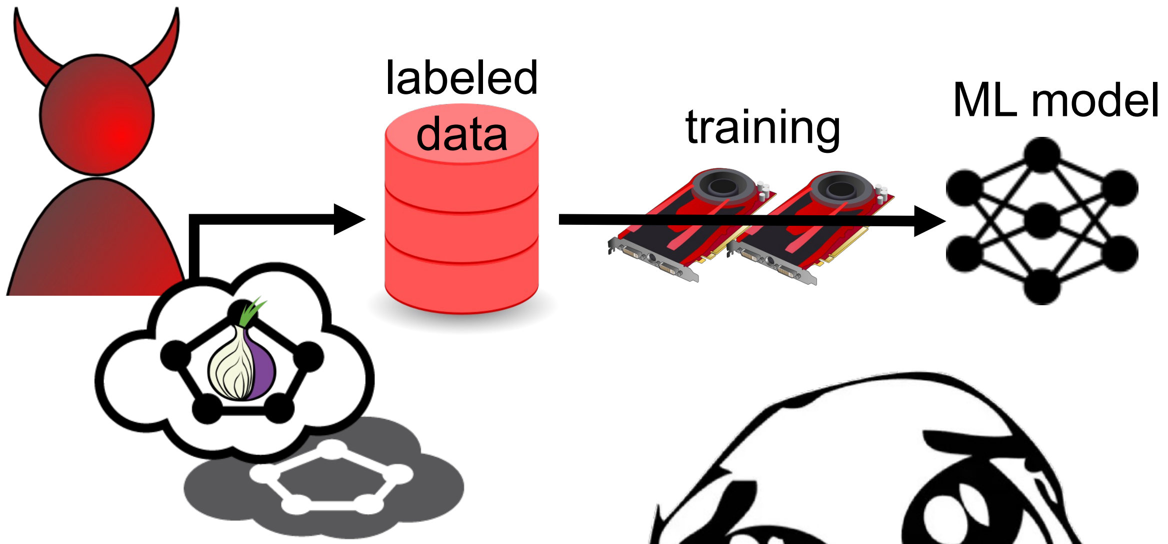




Our Research Direction: Explainable Datasets!

Use network simulation to:

- Increase control over dataset collection
- Augment training with more diverse data
- Explain causal relationships in WF results

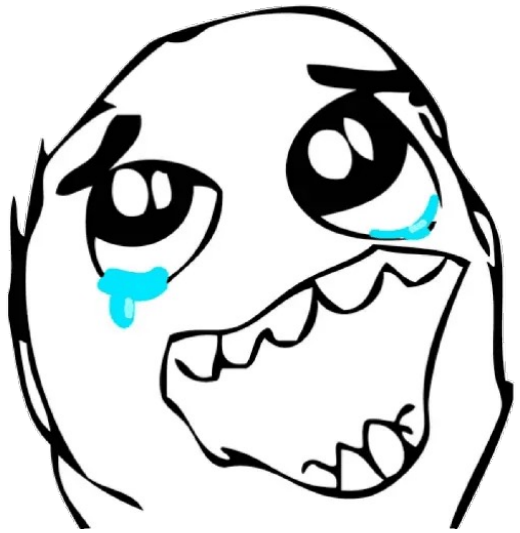


Research Questions:

1. How well can WF attacks be simulated in Shadow?

2. How sensitive is WF to changing network conditions?

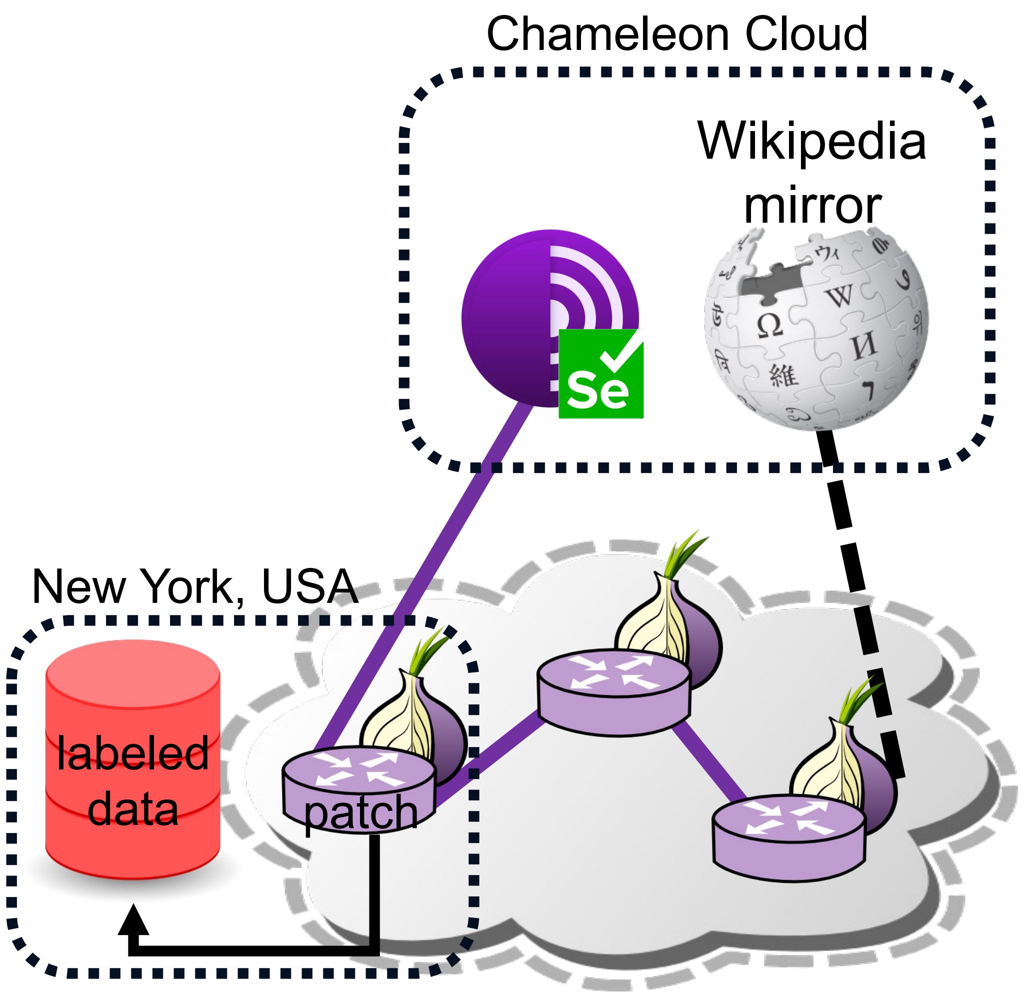
3. How can WF classifiers be made more robust to network effects?



RQ1: How well can WF attacks be simulated in Shadow?

Measurement experiment:
(in both Tor and Shadow)

- Set up Wikipedia mirror (23m pages)
- Choose 98 pages at random
- Fetch each page 200×

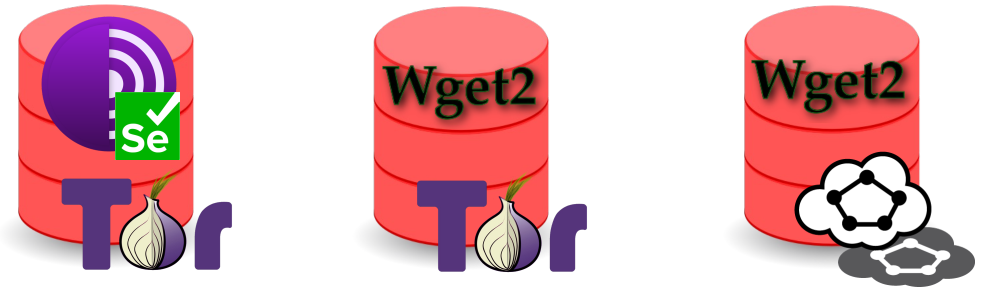


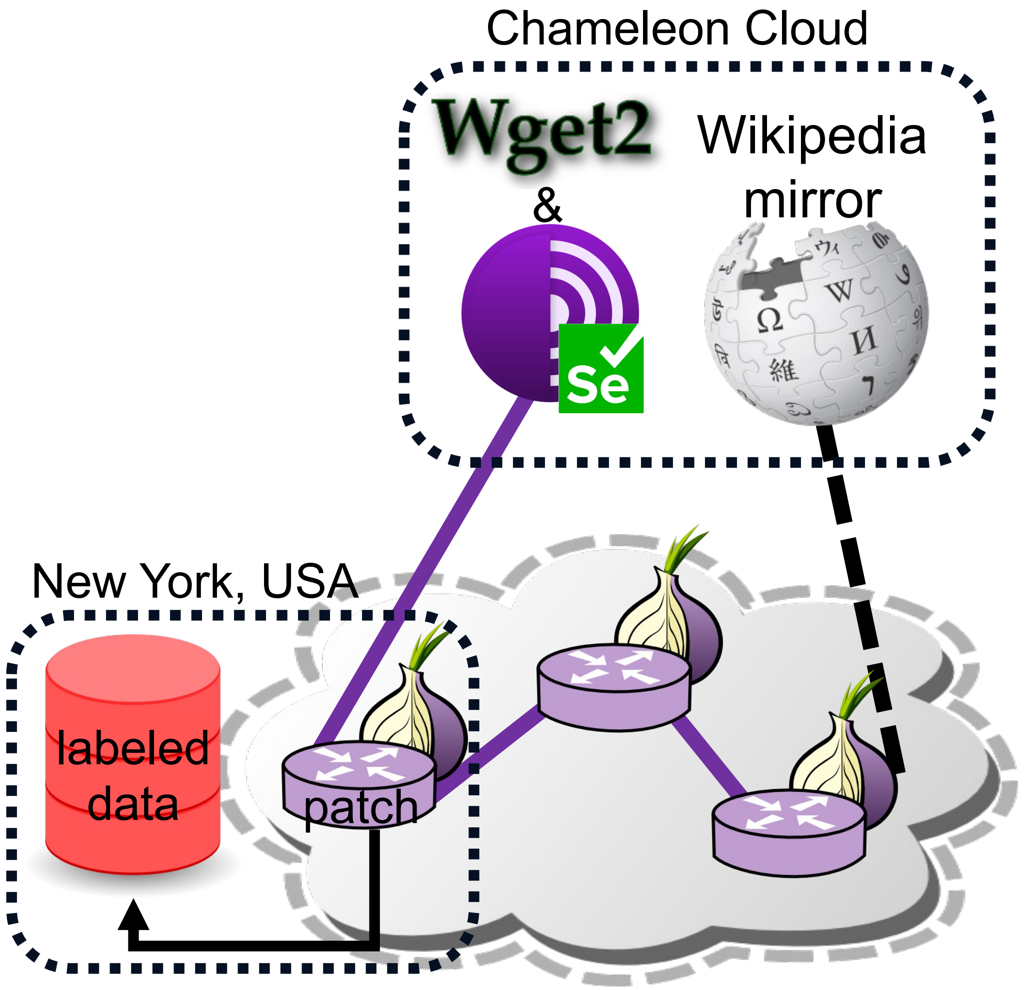
RQ1: How well can WF attacks be simulated in Shadow?

Measurement experiment:
(in both Tor and Shadow)

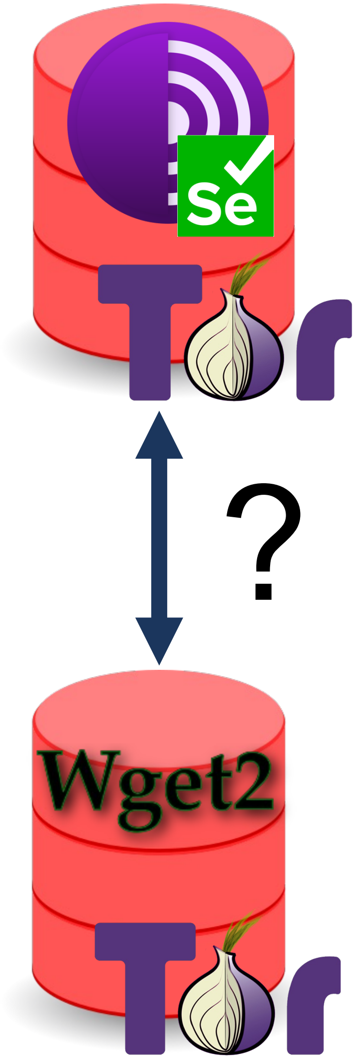
- Set up Wikipedia mirror (23m pages)
- Choose 98 pages at random
- Fetch each page 200×

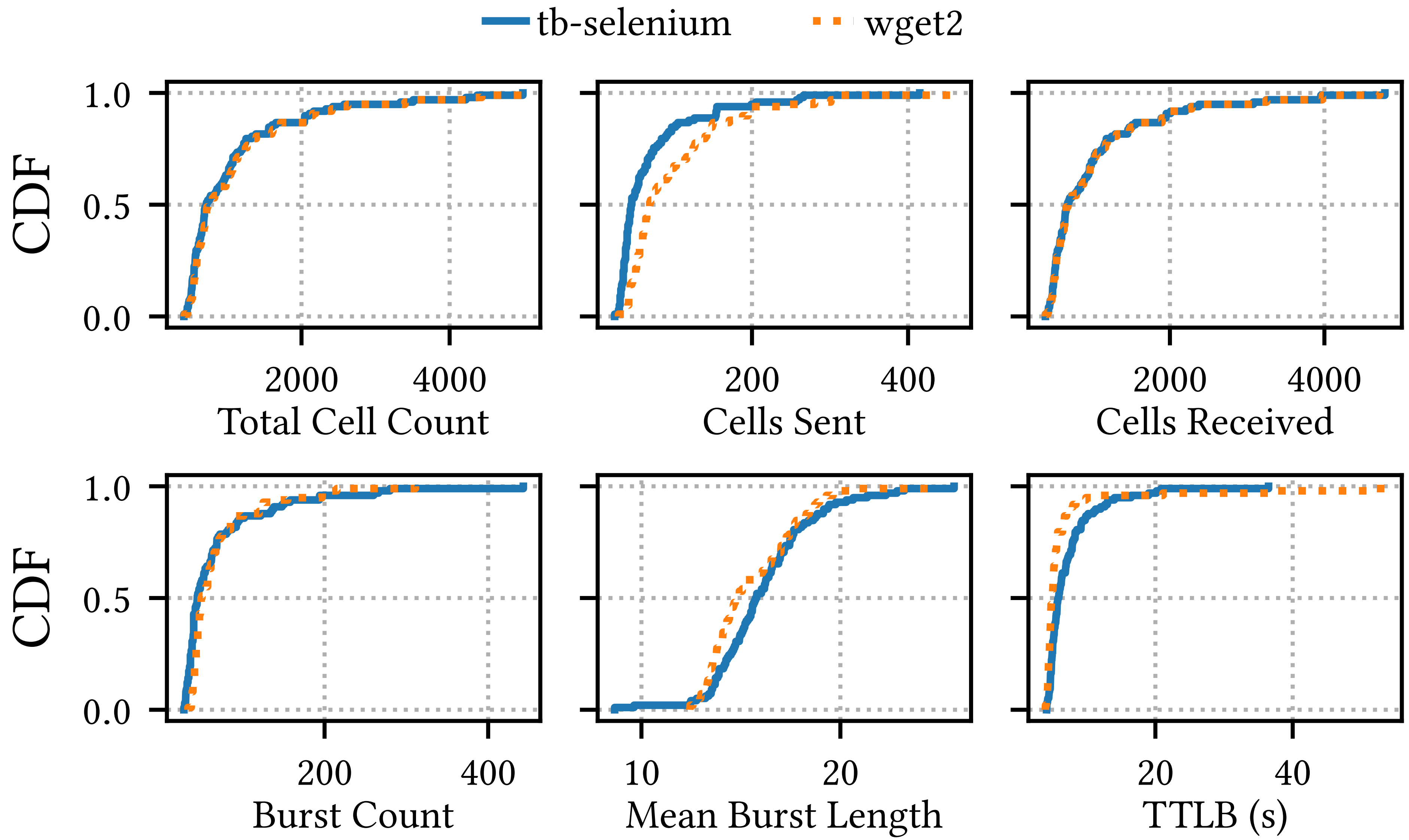
Wrinkle: need to use wget2
(Firefox not yet supported in Shadow)





RQ1: How well can WF attacks be simulated in Shadow?





RQ1: How well can WF attacks be simulated in Shadow?

Classification experiment:

- Multiclass closed world
- 60% train, 40% test (stratified)
- Metric: accuracy (1/98 = random guess)

	Train: Wget2 Tor				Train: Wget2			
Test	CUMUL	*k*-FP	DF	TT	CUMUL	*k*-FP	DF	TT
Wget2 Tor	0.97	0.97	0.97	0.97	0.86	0.83	0.76	0.61
Wget2	0.93	0.88	0.95	0.92	1.0	1.0	1.0	0.99

U.S. NAVAL RESEARCH LABORATORY

RQ1: How well can WF attacks be simulated in Shadow?

Classification experiment:

- Multiclass closed world
- 60% train, 40% test (stratified)
- Metric: accuracy
  (1/98 = random guess)

High accuracy:
(when train-test on same data source)

	Train: Wget2 (Tor)				Train: Wget2 (Shadow)			
Test	CUMUL	*k*-FP	DF	TT	CUMUL	*k*-FP	DF	TT
Wget2 (Tor)	0.97	0.97	0.97	0.97	0.86	0.83	0.76	0.61
Wget2 (Shadow)	0.93	0.88	0.95	0.92	1.0	1.0	1.0	0.99

RQ1: How well can WF attacks be simulated in Shadow?

Classification experiment:

- Multiclass closed world
- 60% train, 40% test (stratified)
- Metric: accuracy
  (1/98 = random guess)

>85% when training completely in simulation!

	Train: Wget2 Tor				Train: Wget2			
	CUMUL	k-FP	DF	TT	CUMUL	k-FP	DF	TT
Test: Wget2 Tor	0.97	0.97	0.97	0.97	0.86	0.83	0.76	0.61
Test: Wget2	0.93	0.88	0.95	0.92	1.0	1.0	1.0	0.99

RQ2: How sensitive is WF to changing network conditions?

Simulation:

- Tor is constantly changing
  - Composition: high relay churn
  - Congestion: variable network usage
- Model with 9 private networks

Composition: re-randomize relays

Congestion: change traffic load

	Seed=1	Seed=2	Seed=3
Low (-25%)			
Baseline			
High (+25%)			

RQ2: How sensitive is WF to changing network conditions?

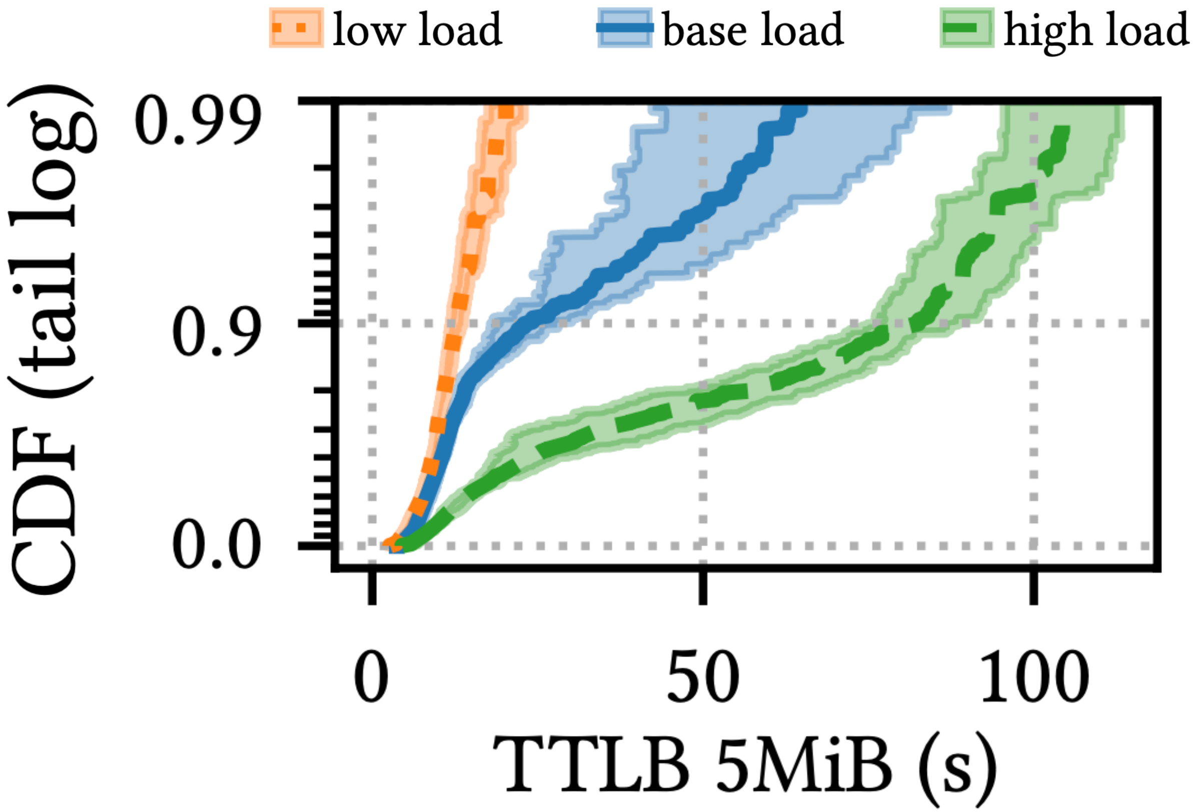
Simulation:

- Tor is constantly changing
  - Composition: high relay churn
  - Congestion: variable network usage
- Model with 9 private networks

Composition: re-randomize relays

Congestion: change traffic load

	Seed=1	Seed=2	Seed=3
Low (-25%)			
Baseline			
High (+25%)			



RQ2: How sensitive is WF to changing network conditions?

Simulation:

- Tor is constantly changing
  - Composition: high relay churn
  - Congestion: variable network usage
- Model with 9 private networks

Datasets:

- Collect webpage traces (Shadow):
  1. Labeled sensitive by adversary (5 pages, 300×)
  2. Benign or unlabeled (30,000×)

Composition: re-randomize relays

		Seed=1	Seed=2	Seed=3
Congestion: change traffic load	Low (-25%)			
	Baseline			
	High (+25%)			

RQ2: How sensitive is WF to changing network conditions?

Simulation:

- Tor is constantly changing
  - Composition: high relay churn
  - Congestion: variable network usage
- Model with 9 private networks

Composition: re-randomize relays

Congestion: change traffic load

	Seed=1	Seed=2	Seed=3
Low (-25%)			
Baseline			
High (+25%)			

Datasets:

- Collect webpage traces (Shadow):
  1. Labeled sensitive by adversary (5 pages, 300×)
  2. Benign or unlabeled (30,000×)

Classification:

- Binary open world (is page sensitive?)
- 60% train, 40% test (stratified)
  - Train 4 classifiers in each of 9 networks
  - Test the 36 classifiers in each network

RQ2: How sensitive is WF to changing network conditions?

Results

1. Variable load had greater effect than variable seed
   - Train low load → test high load particularly poor

TPR	Baseline	Variable Load	Variable Seed
CUMUL	0.99	0.89	0.89
K-FP	0.97	0.78	0.86
DF	0.99	0.89	0.93
TikTok	0.98	0.89	0.93

19 point drop in TPR

RQ2: How sensitive is WF to changing network conditions?

Results

1. Variable load had greater effect than variable seed
   - Train low load → test high load particularly poor
2. Avg. FPR increases more for time-aware classifiers

FPR ($\times 10^{-2}$)	Baseline	Variable Load	Variable Seed
CUMUL	0.165	0.155	0.159
K-FP	0.044	0.237	0.050
DF	0.146	0.290	0.146
TikTok	0.106	0.657	0.123

400-500% increase in FPR

RQ3: How can WF be made more robust to network effects?

Mixture training experiment

- Train using mixture dataset from “training” networks
- Test using examples from independent test network

	Accuracy			TPR			FPR		
	Old	New	%Δ	Old	New	%Δ	Old	New	%Δ
CUMUL	.96	.99	+3	.89	.96	+8	1.55×10^{-3}	1.49×10^{-3}	−4
k-FP	.98	.99	+1	.78	.93	+19	2.37×10^{-3}	5.83×10^{-4}	−75
DF	.98	.99	+1	.89	.95	+7	2.90×10^{-3}	1.49×10^{-3}	−49
TT	.98	.99	+1	.89	.94	+6	6.57×10^{-3}	1.13×10^{-3}	−83

RQ3: How can WF be made more robust to network effects?

Robust classifiers from simulation

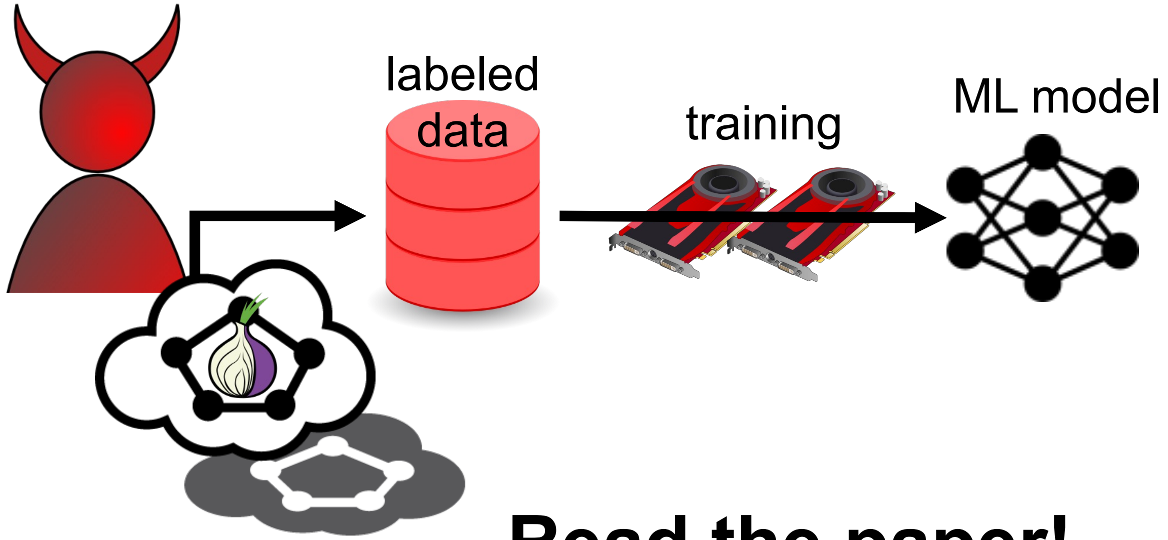
- Use robust mixture training with **100% simulated data**
- Test using the wget2 dataset collected from real-world Tor
- Works well for neural networks, esp. time-aware

	Accuracy			
	CUMUL	*k*-FP	DF	TT
Original	0.86	0.83	0.76	0.61
Robust	0.50	0.70	0.82	0.83
%Δ	-42	-16	+8	+36

Closing

Contact:
rob.g.jansen@nrl.navy.mil
robgjansen.com

<u>Advantages:</u>
- perfectly privacy preserving, no risk/load on Tor
- unlimited source of accurately labeled data
- higher data diversity
- controlled network → explainable data
- simulation-assisted WF outperforms standard methods



Future work:

1. Run Tor Browser directly in Shadow, systematically analyze browser effects
2. Expand analysis beyond Wikipedia
3. Independently useful thrust: study WF using genuine data

Read the paper!