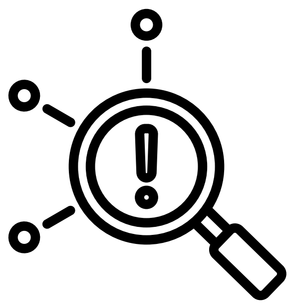
Network monitoring and troubleshooting from within the browser: a data-driven approach

Presented at the RISE Academy research forum

By Sanaa Ghandi, Chadi Barakat and Yassine Hadjadj-Aoul

Why web QoE problems persist ?



Causes

- Network congestion
- Wifi interference
- Device issues
- Network unavailability

For users

- Productivity loss: 70% of users abandon slow pages
- Negative experience: 50% of users report dissatisfaction while streaming

For providers

- Higher churn rate 20–30%
- Increased support costs

How people handle QoE problems today ?

Speedtests (Ookla, fasts.com)	**Intrusive**
Manual check	**Inconsistent**
Wifi intereference	**Time consuming**
Reboot	**Problem specific**

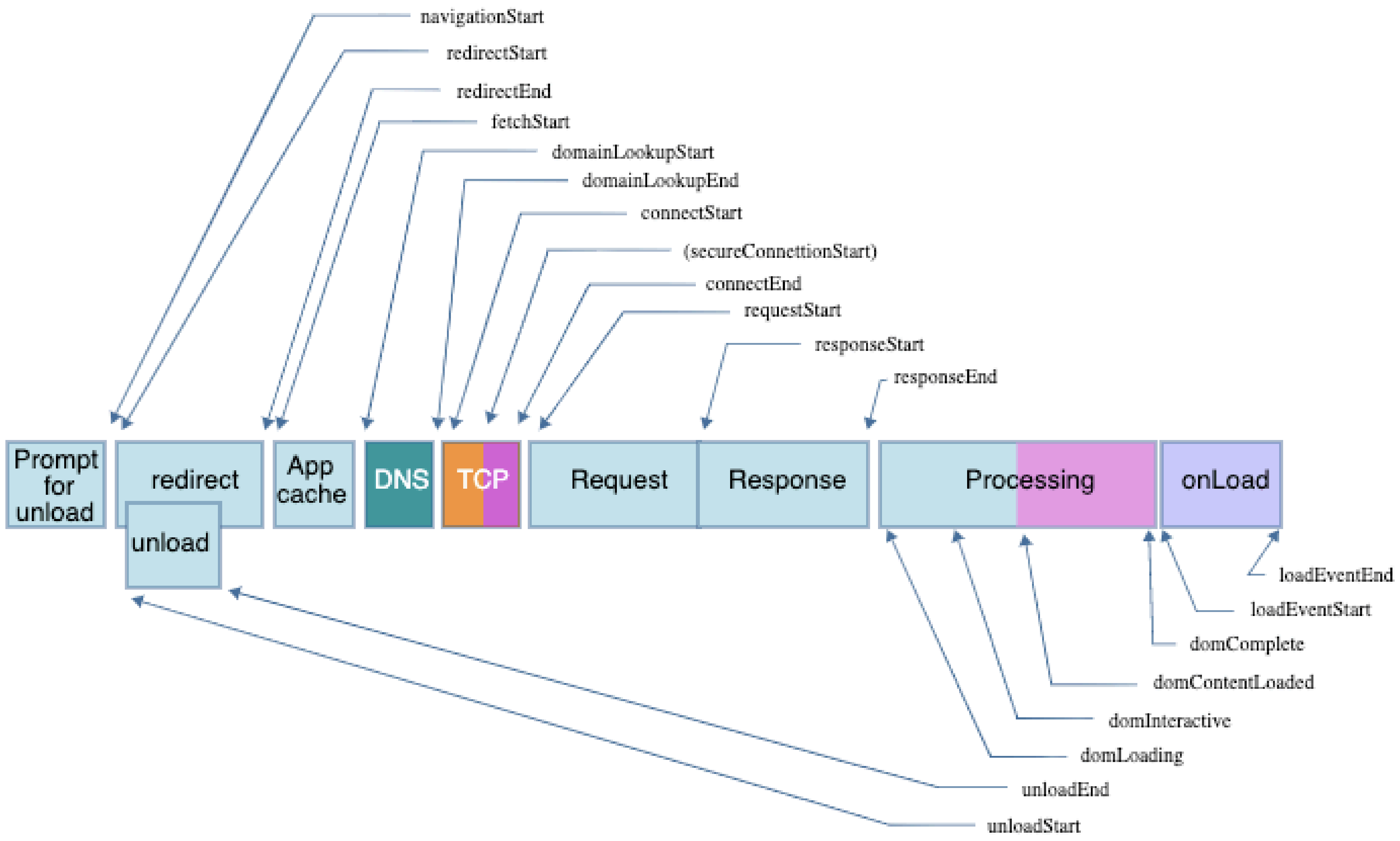
- There is no general solution for network troubleshooting
- The tools are often intrusive and inconsistent
- They are not accessible to everyone
- They can present a high learning curve

The objectives of WEMON

Provide a general lightweight network troubleshooting solution :

- **Automated**
- **Non intrusive**
- **Lightweight**
- **Within the browser**
- **Fast**
- **User friendly**

Can data within the browser be valuable ?



Our assumption is that browser contains a wealth of data freely available

- **Performance API:** rendering data
- **Network information API:** connection type
- **Chrome API:** CPU usage, memory usage

- **Does this data contain network problem signature ?**
- **Would it help to troubleshoot network problems ?**

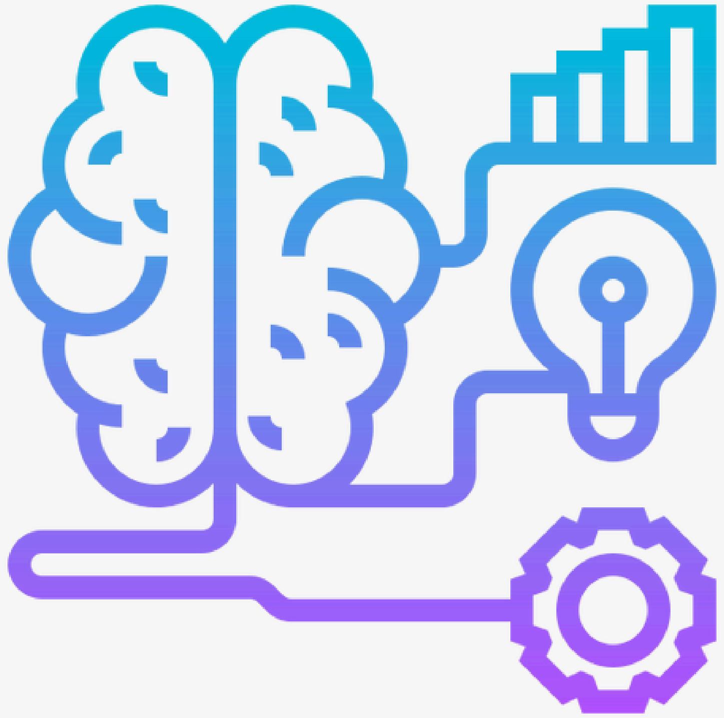
Reference: Navigation and resource timings
Url: https://developer.mozilla.org/en-US/docs/Web/Performance/Navigation_and_resource_timings

Our data-driven approach

BROWSER DATA AS INPUT

RENDERING	pageloadtime, ttfb, redirect, dns, connect, request, response, dom, domParse, domScripts, contentLoaded, domSubRes, load, RUMSpeedIndex
CPU	cpuName, cpuArch, cpuFeatures, cpuNumOfProcessors, systemCapacity, systemAvailableCapacity, delta_idle, delta_kernel, delta_total, delta_user
NETWORK	effectiveType downlink rtt protocol

MACHINE LEARNING



OUTPUT

- **Network performance metrics** (Bandwidth, packet loss, RTT…)
- **Root cause analysis and troubleshooting**
- **Embedded in a browser extension**

Dataset building by controlled experimentation

1.Controlled experimentation

2.Data collection from the browser

3.Ground truth measurement with specialized tools

4.Model calibration and training

5.Iterative improvement

- Browsing under various network conditions
- Creating various datasets

- Collecting data using an extension

- Establishing accurate labels and benchmarks

- Choosing the model
- Finding the optimal parameters

Challenges

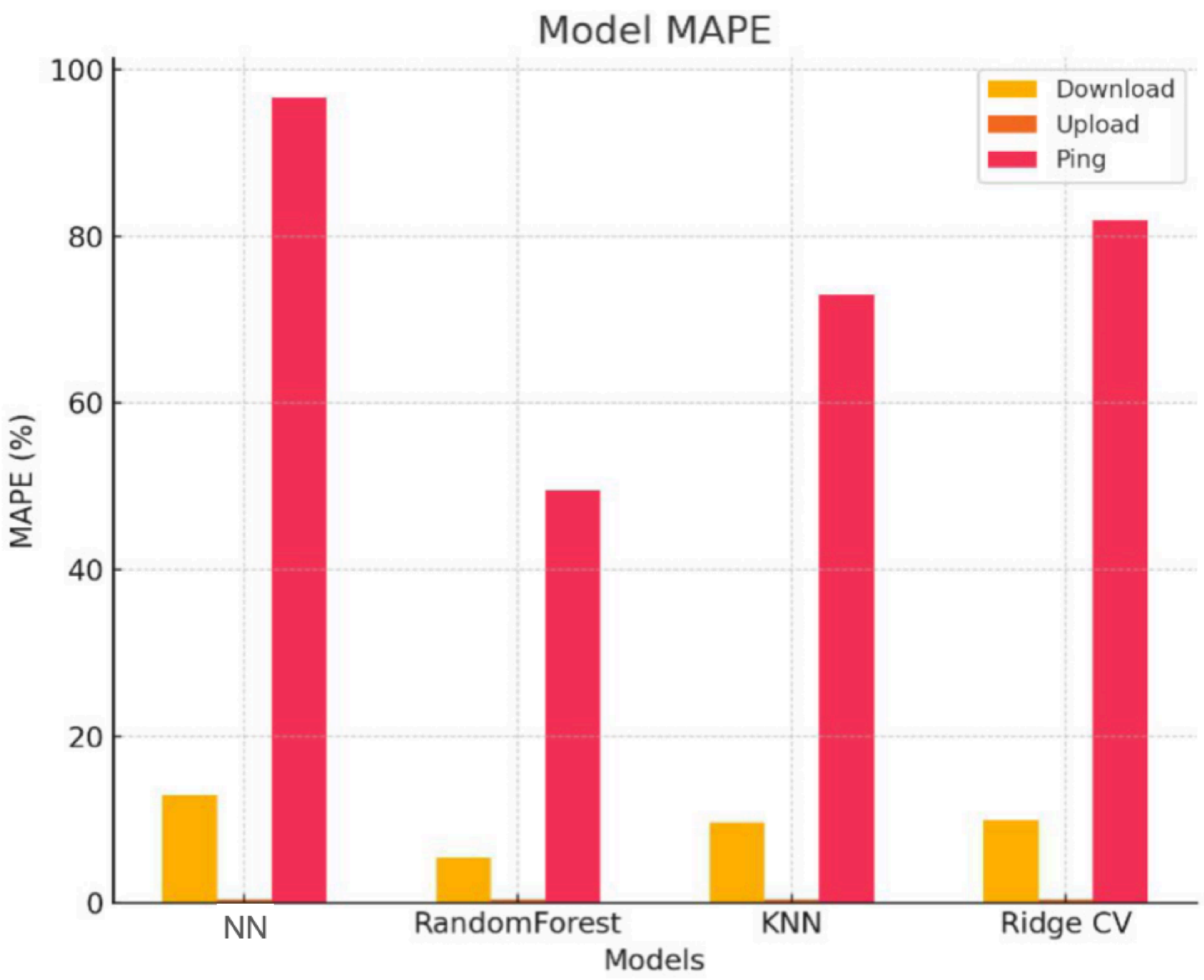
- Setting up the experimental platform
- Definition of scenarios
- Traffic generation
- Collection and labelling of the dataset
- Analysis of the accuracy and robustness of the ML models

Speedtest–like inference without packet injection

- **Goal:** predict latency, uplink and downlink bandwidth like a speedtest tool
- **Methodology:**
  - Controlled experiments in the wild
  - Real wifi conditions
  - Control and choice of the pages from Hispar [Aqeel and al. IMC'20]
  - 3400 data samples
  - Ground-truth obtained with real Ookla Speedtests

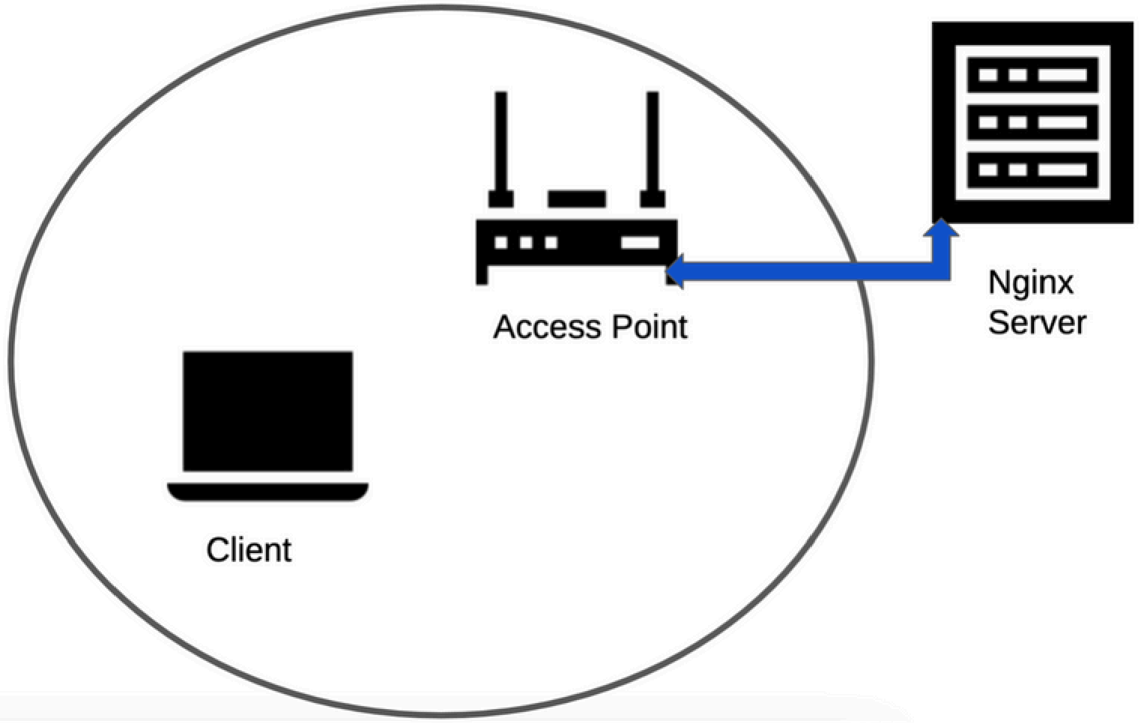
Results

- Mean absolute percentage error (MAPE)
- Best performance for random forest
- No need for more complex models



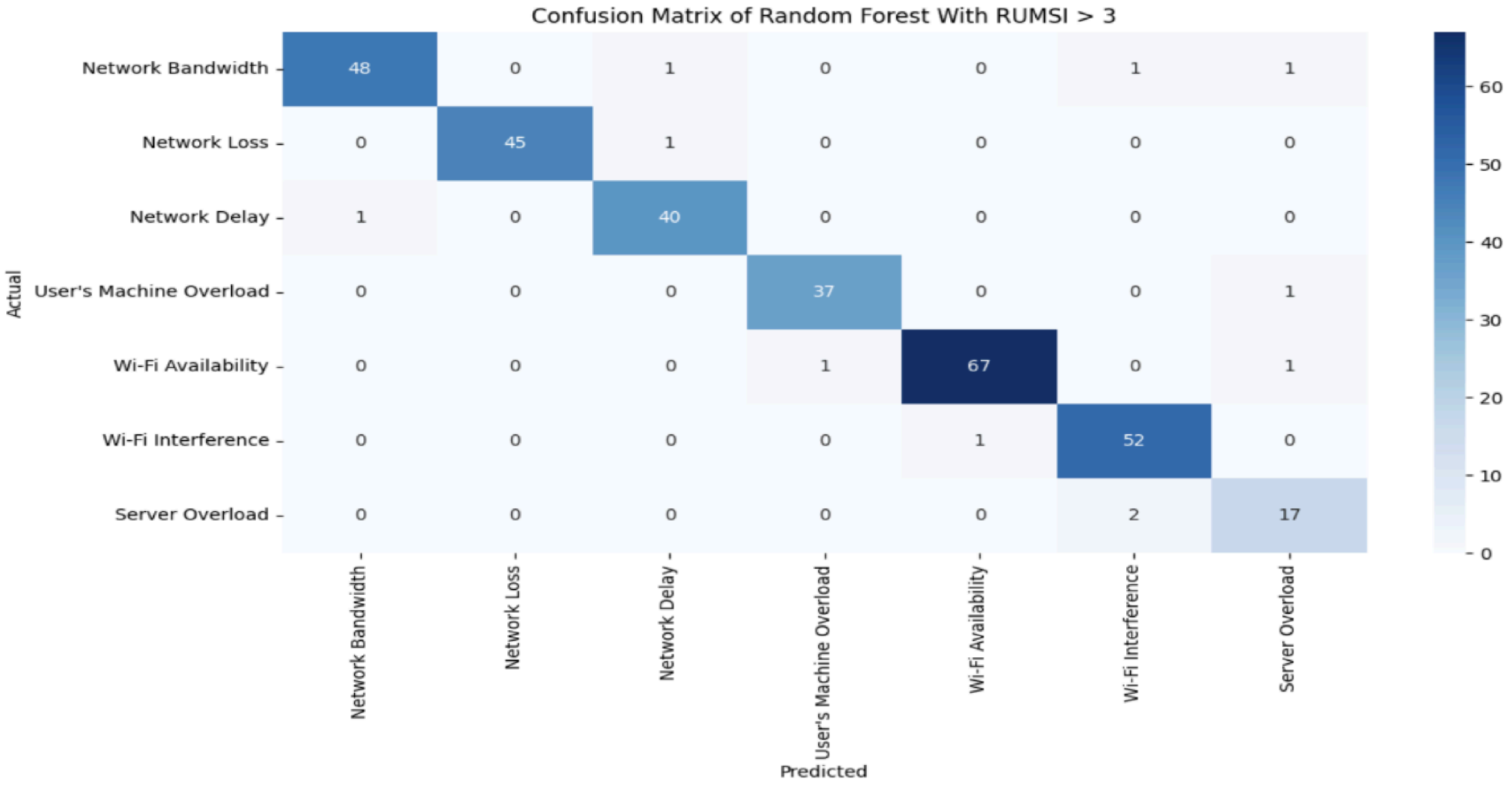
Network troubleshooting

- **Goal:** inferring the origin of slow web browsing
- **Methodology:**
  - Mininet wifi
    - Consideration of 7 types of network anomalies (*Wi-Fi medium availability, Wi-Fi interference, network bandwidth, packet loss, delay, user machine overload, server overload*)
    - One anomaly at a time
  - 33 total scenarios
  - 100 web pages replicated locally



Results

- Model accuracy: 96,53%
- Results published in IWCMC 2024 [1]



[1] Passive network monitoring and troubleshooting from within the browser: a data-driven approach, Naomi Kirimi, Chadi Barakat, Yassine Hadjadj-Aoul

Ongoing work

- Stressing the methodology in more realistic scenarios
  - Multiple anomalies at a time
  - 43k Network performance scenarios
  - 45k Variable web pages
- Integration of performance estimation and troubleshooting
- Exploration of the methodology in video streaming context

Thank you for your attention

sanaa.ghandi@inria.fr