# Exploring Browser Fingerprinting

4 collaborators

# 00
# Overview

Analyzing the differences in fingerprinting techniques used by different browsers and industries

# 01

# FPMON Extension

A real-time fingerprinting monitor as a browser extension

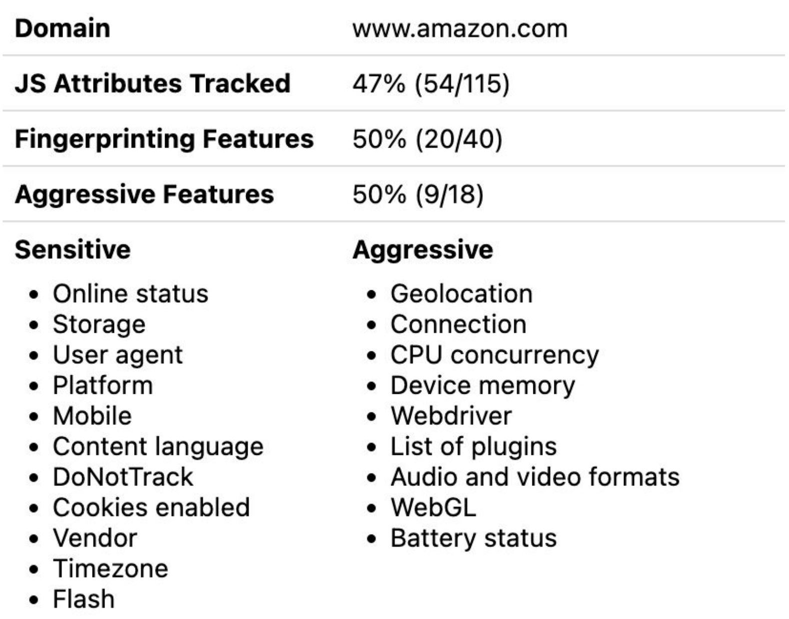
# FPMON: Real-time Browser Fingerprinting Monitor

**Browser**
Extension

- Developed as part of a large-scale study in 2020
- Provides the user with real-time feedback on what browser fingerprinting method is being applied against them

# How does it work?

- FPMON monitors all JavaScript functions
- The fingerprinting techniques (FP) are then grouped and returned to the user
- By tracking the domain accessed and the FP techniques used, we can record this data and look into modeling the different domains and how user's data is accessed
- We will further look into the different industries behind each domain



| **Domain** | www.amazon.com |
|---|---|
| **JS Attributes Tracked** | 47% (54/115) |
| **Fingerprinting Features** | 50% (20/40) |
| **Aggressive Features** | 50% (9/18) |

**Sensitive**

- Online status
- Storage
- User agent
- Platform
- Mobile
- Content language
- DoNotTrack
- Cookies enabled
- Vendor
- Timezone
- Flash

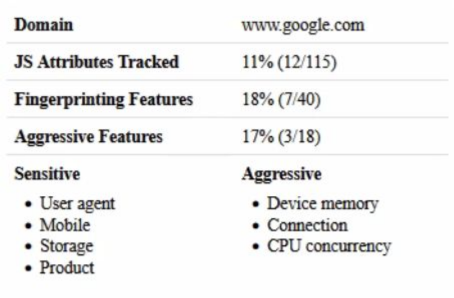
**Aggressive**

- Geolocation
- Connection
- CPU concurrency
- Device memory
- Webdriver
- List of plugins
- Audio and video formats
- WebGL
- Battery status

# FPMON Demo

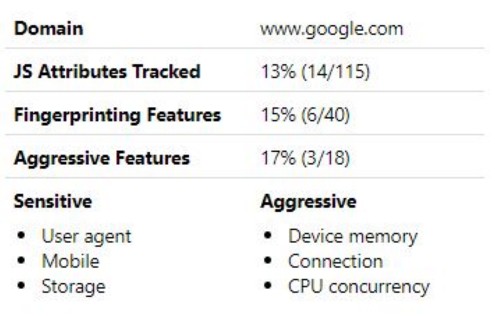
FPMON can be used as an extension on both Chrome and Firefox, the two browsers we were looking at for the next step in this project.

## Firefox + Search Browser



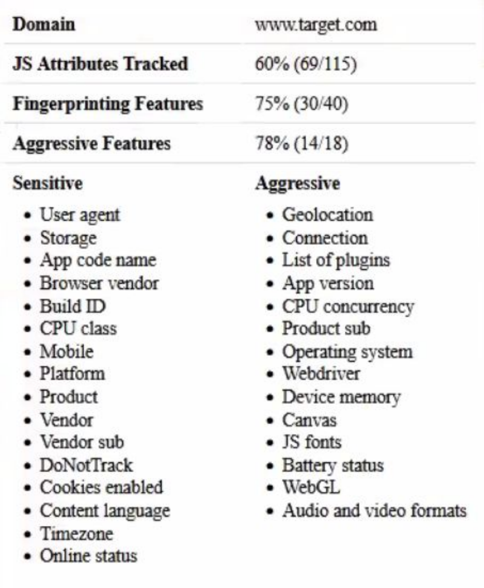
| | |
|---|---|
| **Domain** | www.google.com |
| **JS Attributes Tracked** | 11% (12/115) |
| **Fingerprinting Features** | 18% (7/40) |
| **Aggressive Features** | 17% (3/18) |
| **Sensitive** | **Aggressive** |
| • User agent<br>• Mobile<br>• Storage<br>• Product | • Device memory<br>• Connection<br>• CPU concurrency |

## Chrome + Search Browser



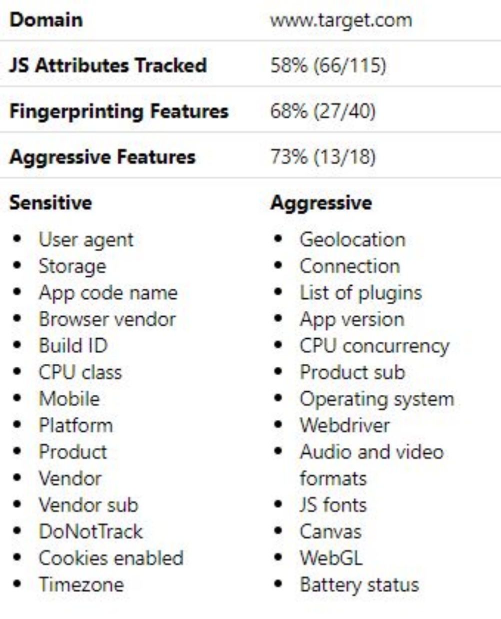
| | |
|---|---|
| **Domain** | www.google.com |
| **JS Attributes Tracked** | 13% (14/115) |
| **Fingerprinting Features** | 15% (6/40) |
| **Aggressive Features** | 17% (3/18) |
| **Sensitive** | **Aggressive** |
| • User agent<br>• Mobile<br>• Storage | • Device memory<br>• Connection<br>• CPU concurrency |

## Firefox + Retail



| | |
|---|---|
| **Domain** | www.target.com |
| **JS Attributes Tracked** | 60% (69/115) |
| **Fingerprinting Features** | 75% (30/40) |
| **Aggressive Features** | 78% (14/18) |
| **Sensitive** | **Aggressive** |
| • User agent<br>• Storage<br>• App code name<br>• Browser vendor<br>• Build ID<br>• CPU class<br>• Mobile<br>• Platform<br>• Product<br>• Vendor<br>• Vendor sub<br>• DoNotTrack<br>• Cookies enabled<br>• Content language<br>• Timezone<br>• Online status | • Geolocation<br>• Connection<br>• List of plugins<br>• App version<br>• CPU concurrency<br>• Product sub<br>• Operating system<br>• Webdriver<br>• Device memory<br>• Canvas<br>• JS fonts<br>• Battery status<br>• WebGL<br>• Audio and video formats |

## Chrome + Retail



| | |
|---|---|
| **Domain** | www.target.com |
| **JS Attributes Tracked** | 58% (66/115) |
| **Fingerprinting Features** | 68% (27/40) |
| **Aggressive Features** | 73% (13/18) |
| **Sensitive** | **Aggressive** |
| • User agent<br>• Storage<br>• App code name<br>• Browser vendor<br>• Build ID<br>• CPU class<br>• Mobile<br>• Platform<br>• Product<br>• Vendor<br>• Vendor sub<br>• DoNotTrack<br>• Cookies enabled<br>• Timezone | • Geolocation<br>• Connection<br>• List of plugins<br>• App version<br>• CPU concurrency<br>• Product sub<br>• Operating system<br>• Webdriver<br>• Audio and video formats<br>• JS fonts<br>• Canvas<br>• WebGL<br>• Battery status |

# 02

# Predictive Machine Learning Models

# Dataset and Features

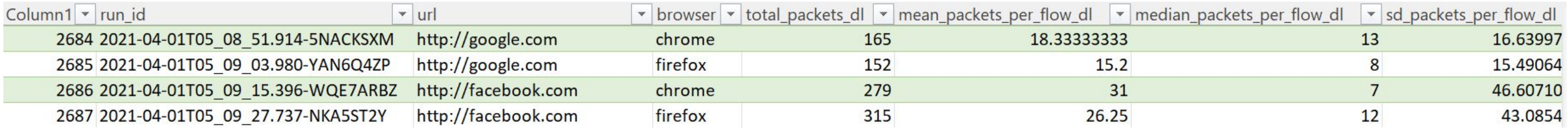
## Dataset Features

- Various Statistics
    - Max, min, sd, mean, median
- Packet
- Flow
- Packet Size
- Burst
- Duration
- Entropy
- Time Between Flows

## Notable Features

- Url
- Packets in downlink
- Packets  in uplink

*Browser Fingerprinting: How to Protect Machine Learning Models and Data with Differential Privacy?*
By Dietz, Muhlhauser, Seufert, Gray, Hoßfeld, Herrmann



| Column1 | run_id | url | browser | total_packets_dl | mean_packets_per_flow_dl | median_packets_per_flow_dl | sd_packets_per_flow_dl |
|---|---|---|---|---|---|---|---|
| 2684 | 2021-04-01T05_08_51.914-5NACKSXM | http://google.com | chrome | 165 | 18.33333333 | 13 | 16.63997 |
| 2685 | 2021-04-01T05_09_03.980-YAN6Q4ZP | http://google.com | firefox | 152 | 15.2 | 8 | 15.49064 |
| 2686 | 2021-04-01T05_09_15.396-WQE7ARBZ | http://facebook.com | chrome | 279 | 31 | 7 | 46.60710 |
| 2687 | 2021-04-01T05_09_27.737-NKA5ST2Y | http://facebook.com | firefox | 315 | 26.25 | 12 | 43.0854 |

# Data Processing

## Y

- Browser
  - Chrome
  - Firefox

## Processing

- Removal of Null Values
- StandardScaler
- OneHotEncoding
- OrdinalEncoding

4495 x 146 (col x rows)

# Models

| Model | RMSE |
|---|---|
| Linear Regression | 63.419 |
| Random Forest | 0.128 |
| Logistic Regression | 0.192 |
| Ridge Regressor | 94.746 |
| Random Bagging Regressor | 0.242 |

03

# Future Works

What we hope to look more into

# End goal

How do browser fingerprinting techniques differ across various industries?

# Outline

### Categorize given data by industry

Separate both FPMON and networking dataset into industries in which the company in the url falls under

### Gather data on selected urls

Choose a certain number of urls present in the dataset that falls under each industry and gather the relevant data

### Run ML models to on FPMON and networking datasets

Predict the industry based on fingerprinting techniques/data

### Analyze findings

Analyze how browser fingerprinting techniques differ among websites of different industries

# 04

# Thank you!

Any questions?