AdGraph: A Machine Learning Approach to Automatic and Effective Adblocking

From Arxiv

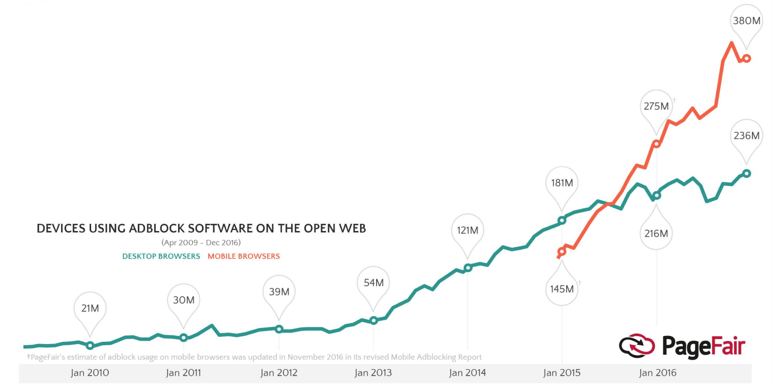
Security Reading Group -- 11/28

Outline

- Background
- Motivation
- Approach
  - System overview
  - Graph representation
  - ML model
  - Challenges
- Evaluation
  - Accuracy
  - Against adversaries
- Conclusions

Background

- Adblockers are used on more than 600 million devices globally as of December 2016
- Adblockers use manually curated filter lists to block ads and trackers based on informally crowdsourced feedback from the adblocking community
- State-of-the-art: manually curated filter lists with RegExp-based rules

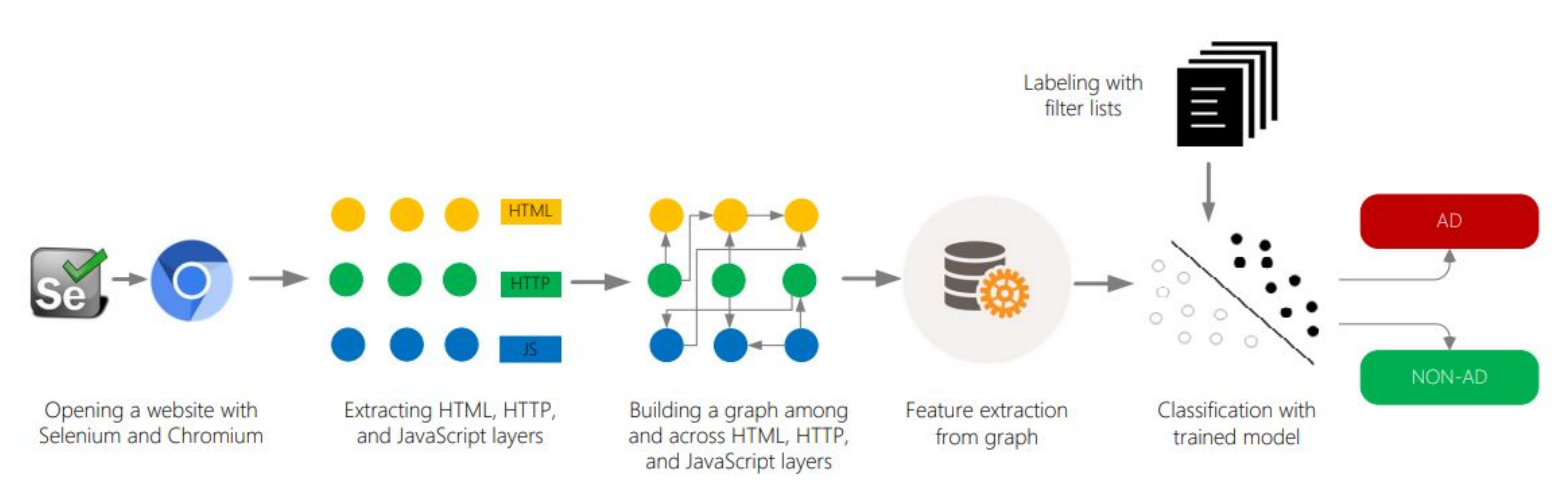


List	# Rules
EasyList	72,660
EasyPrivacy	15,507
Anti-Adblock Killer	1,964
Warning Removal List	378
Blockzilla	1,155
Fanboy Annoyances List	38,675
Peter Lowe's List	2,962
Squid Blacklist	4,485

Motivation

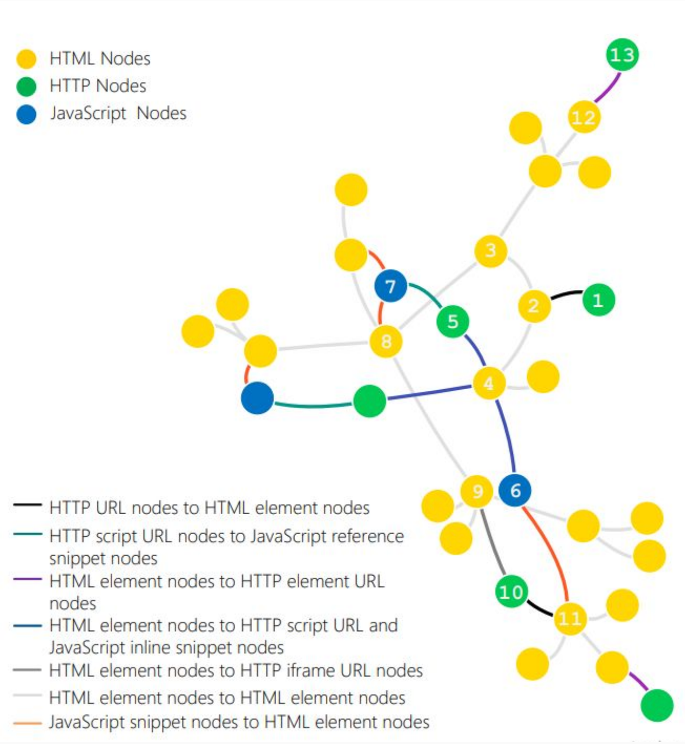
- **Limitations of state-of-the-art (manual lists)**
  - **Manual nature is problematic**
    - **Bloat (FN, less than 3% HTTP rules in EasyList trigger)**
    - **Accuracy (FP, exception rules catering for other overly broad rules)**
  - **Evasion from publishers**
    - **Concealing signatures**
      - **Obfuscation (native ads etc.)**
      - **Domain Generation Algorithm**
    - **Anti-adblockers**
      - **Actively detects adblockers and issues warning messages**
- **Some ML attempts to automate filter list curation, which are mostly based on URL features only**
- **Root weaknesses:**
  - **manual** process;
  - **contextless/hardcoded** rules

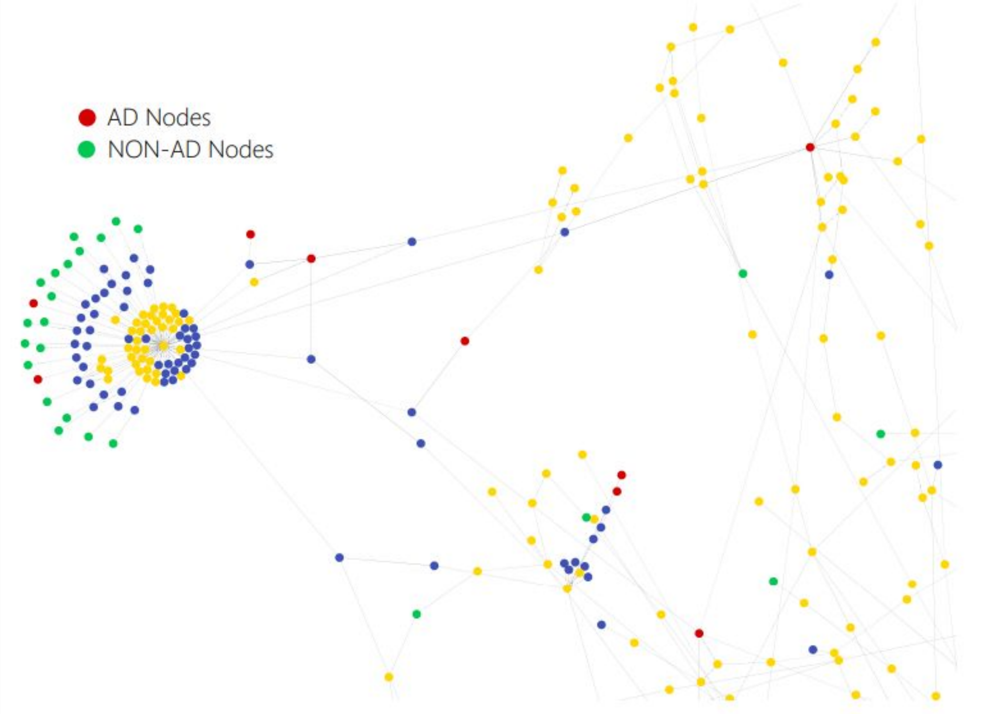
AdGraph - overview



- **Automatic:** ML classification based on existing filter lists
- **Contextual/structural:** combining HTML/HTTP/JS layers' information into a graph

AdGraph - graph representation

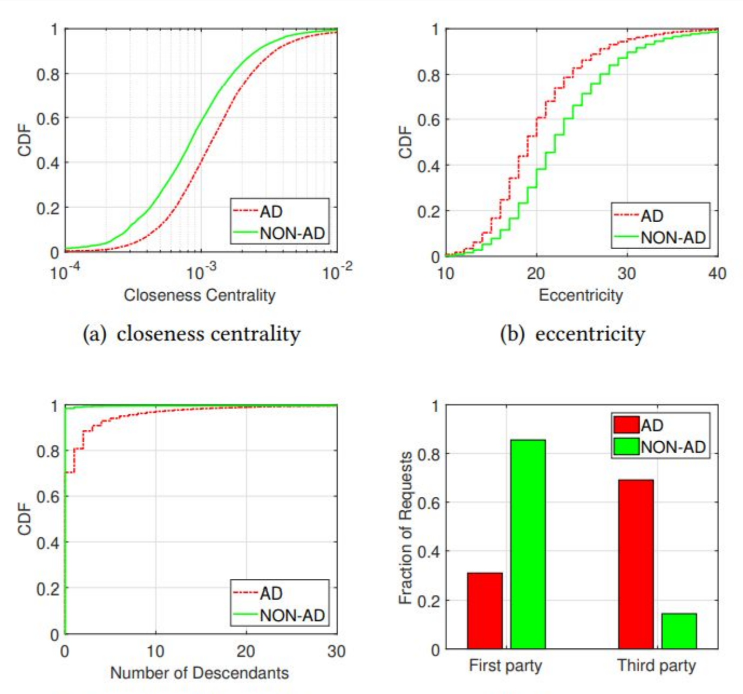




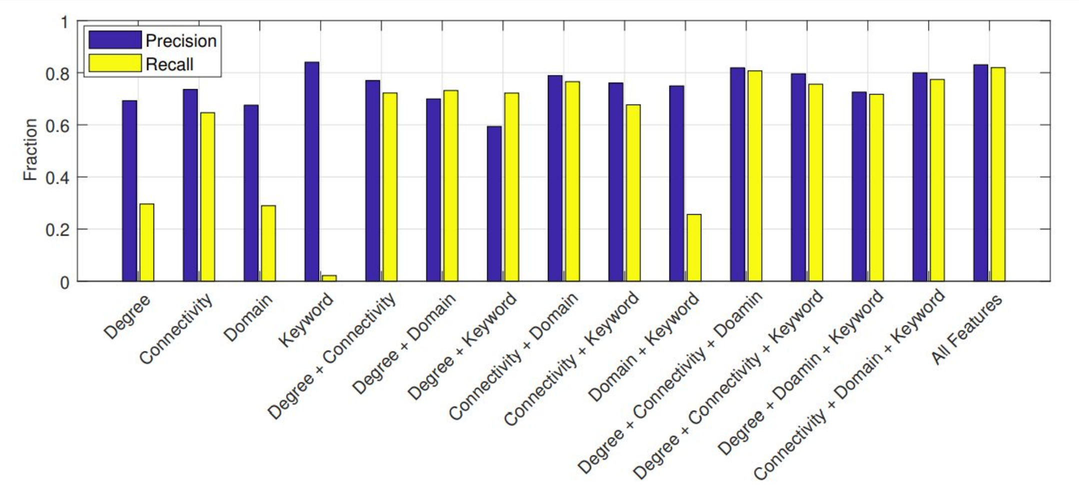
AdGraph - ML model

- **Random forest - an ensemble of 10 decision trees**
- **Features:**
  - **Degree Features**
    - In-Degree/Out-Degree/Descendants/Addition of nodes/Modification of node attributes/Event listener attachment
  - **Connectivity Features**
    - Katz centrality/Closeness centrality/Mean degree connectivity/Eccentricity
  - **Domain Features**
    - Domain party/Sub-domain/Base domain in query string/Same base domain and request domain/Node category
  - **Keyword Features**
    - Ad keywords/Query string parameters/Ad dimension information in query string

AdGraph - ML model



(a) closeness centrality
(b) eccentricity
(c) number of descendants
(d) domain party



AdGraph - challenges

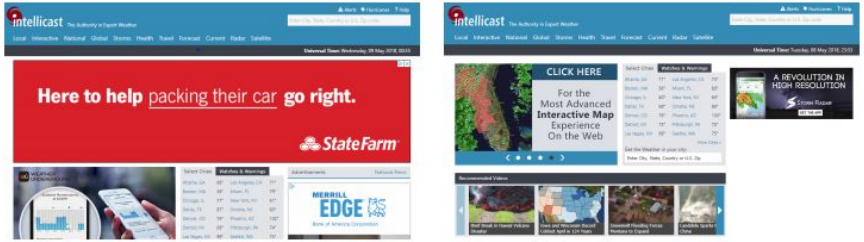
- **Most of same- and cross-layer links in graph are straightforward to establish**
- **JS attribution is tricky - how to track which JS created/modified this HTML element**
  - Browser instrumentation required, no existing mechanism
  - We use JSgraph [1], a forensics tool originally designed for monitoring JS activities
    - It leverages the single-threaded nature of JS engine (i.e. at any given time point there can be only one JS being executed)
    - It instruments points when control is exchanged between Blink and V8 (i.e. `createElement()` etc.), and attributes the event to the executing script
    - We sync all nodes in the same page to construct the graph

Evaluation - accuracy

- **10-fold cross-validation among Alexa Top 10K crawl: 97.7% accuracy, 83.0% precision, and 81.9% recall**
- **We have disagreements with filter lists, but are they really our mistakes, or the lists are wrong?**
    - **We did FP analysis to confirm a sample of these disagreements**

Functional	Advertising/Tracking	N/A	Unknown
427 (30.5%)	915 (65.4%)	23 (1.6%)	35 (2.5%)

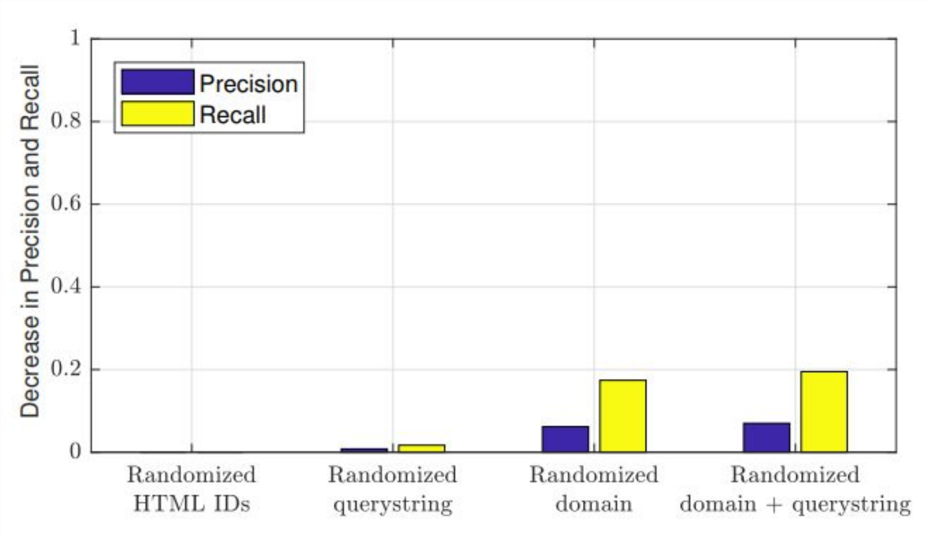
Table 1: Breakdown of false positive analysis



(a) Before blocking (b) After blocking

Evaluation - adversary

- **HTML Element Obfuscation**
- **HTTP URL Obfuscation**
  - Query string randomization
  - Domain name randomization
  - Randomization of both query string and domain name



Conclusions

- Graph-based machine learning approach that automatically and effectively block ads and trackers on the web
- Replicates popular crowdsourced filter lists with an 97.7% accuracy
- Detects a significant number of ads and tracker which are missed by popular crowdsourced filter lists
- Applications
  - Offline use
    - Improving filter lists
    - Automatic ad blocking in less critical regions
  - Online use (WIP)
    - Live in-browser ad blocking

Q & A?

https://arxiv.org/abs/1805.09155