Automatically Detecting Vulnerable Sites Before They Turn Malicious

Kyle Soska
Carnegie Mellon University
ECE / Cylab
ksoska@cmu.edu

Nicolas Christin
Carnegie Mellon University
ECE / Cylab
nicolasc@cmu.edu

Sponsors: NSF (CCF-0424422, CNS-1223762) and DHS S&T/CSD (N66001-13-C-0131)

Problem Setting

- Adversaries compromise websites
  - Economically Rational – monetize compromises
    - Neutral to victims
    - Maximize volume, efficiency, profits
  - Hacktivist – promote social, political, or religious agenda
    - Targeted attacks
    - Low volume

Economically Rational Adversary

- Decisions always maximize profit
- Probabilistic Polynomial Time
  - Cannot break standard crypto, session cookies, hashes, etc.
- Does not control significant portion of web
  - Cannot perform adversarial machine learning attacks by poisoning a random sample of the web
- Able to exploit vulnerable web software

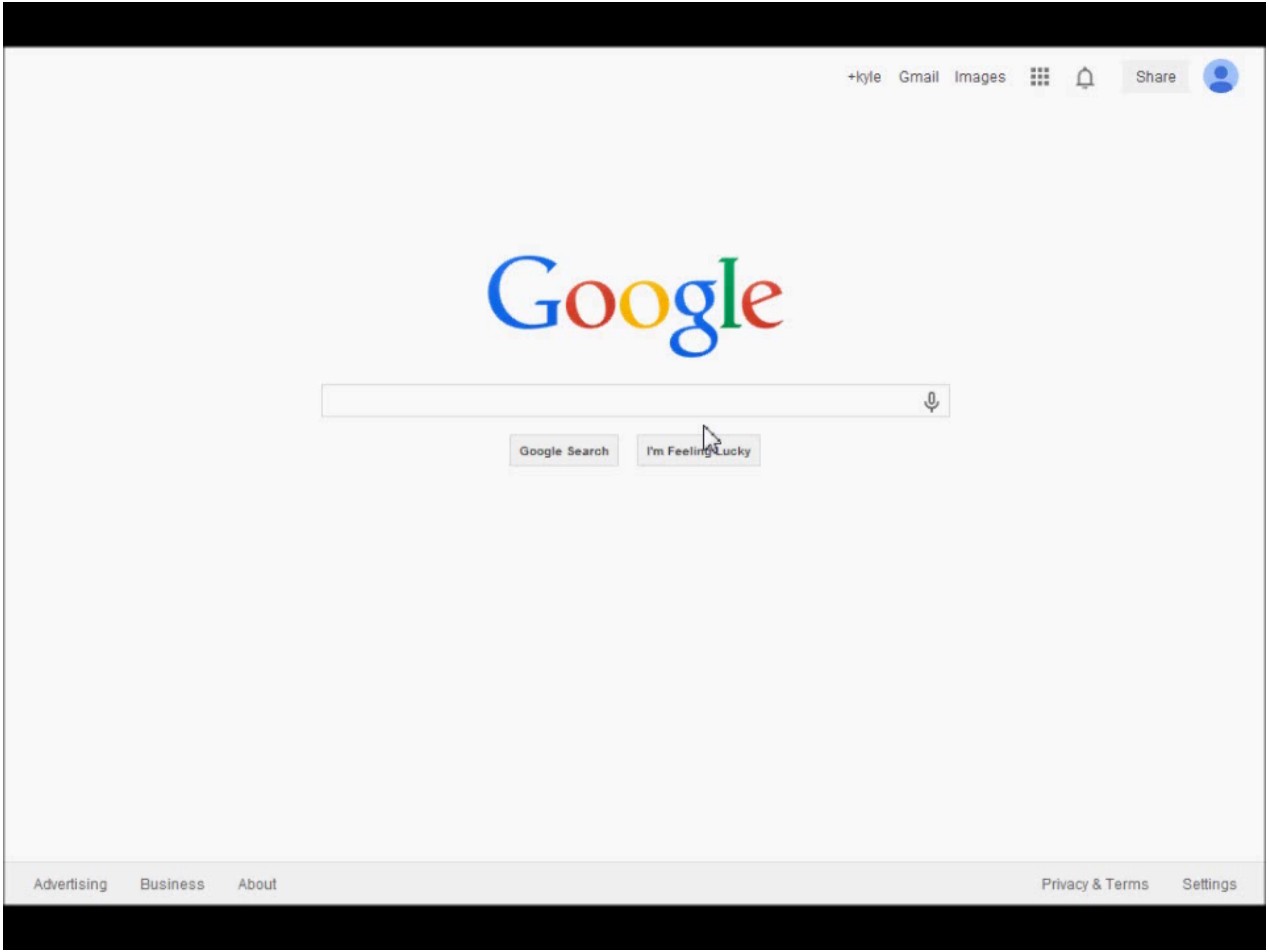
Mode of Operation

Step 1: Find bug or vulnerability in popular web software or content management system (CMS)

Step 2: Enumerate sites containing vulnerability

Step 3: Exploit vulnerable sites

Step 4: Monetize and profit

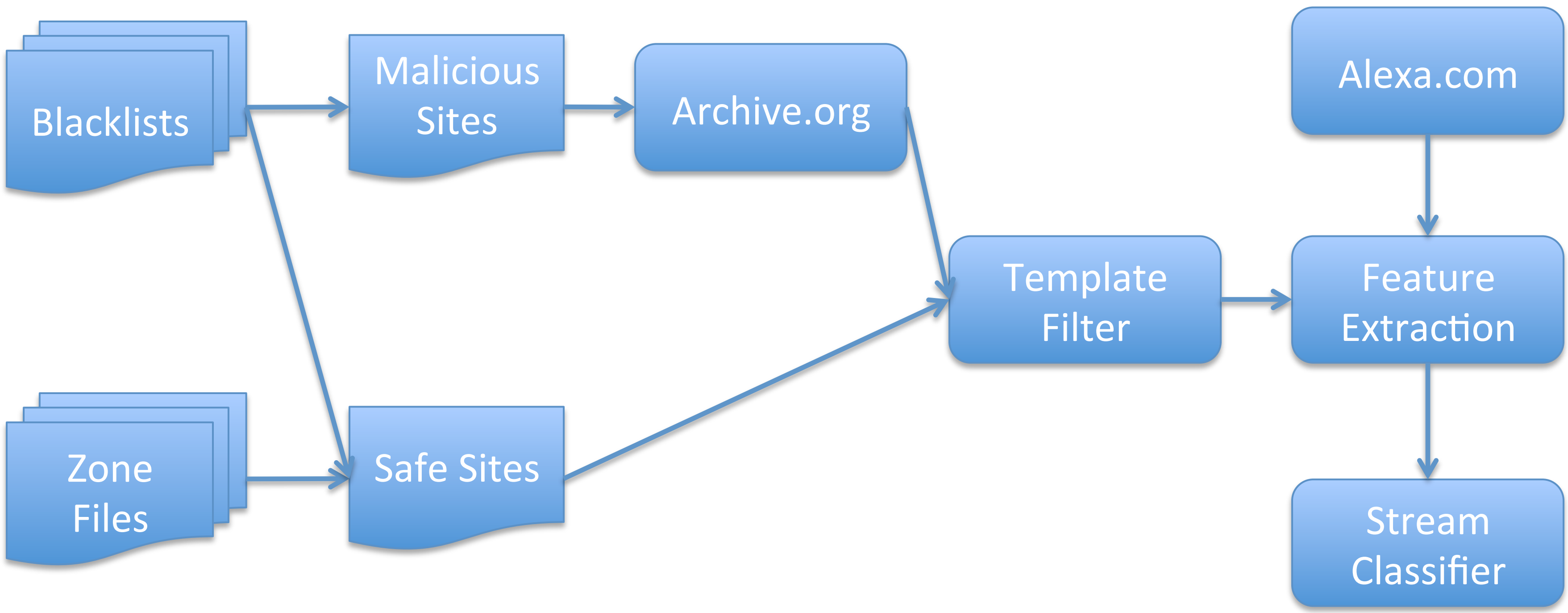

+kyle
Gmail
Images
Share
Google
Google Search
I'm Feeling Lucky
Advertising
Business
About
Privacy & Terms
Settings

Problem and Goal

- Existing approaches detect if a **webpage** is **already** malicious

- Is it possible to predict if a non-malicious **website** will become malicious in the **future**?
  - What would such a system look like?
  - What requirements are imposed on such a system?
  - What are the fundamental limitations?

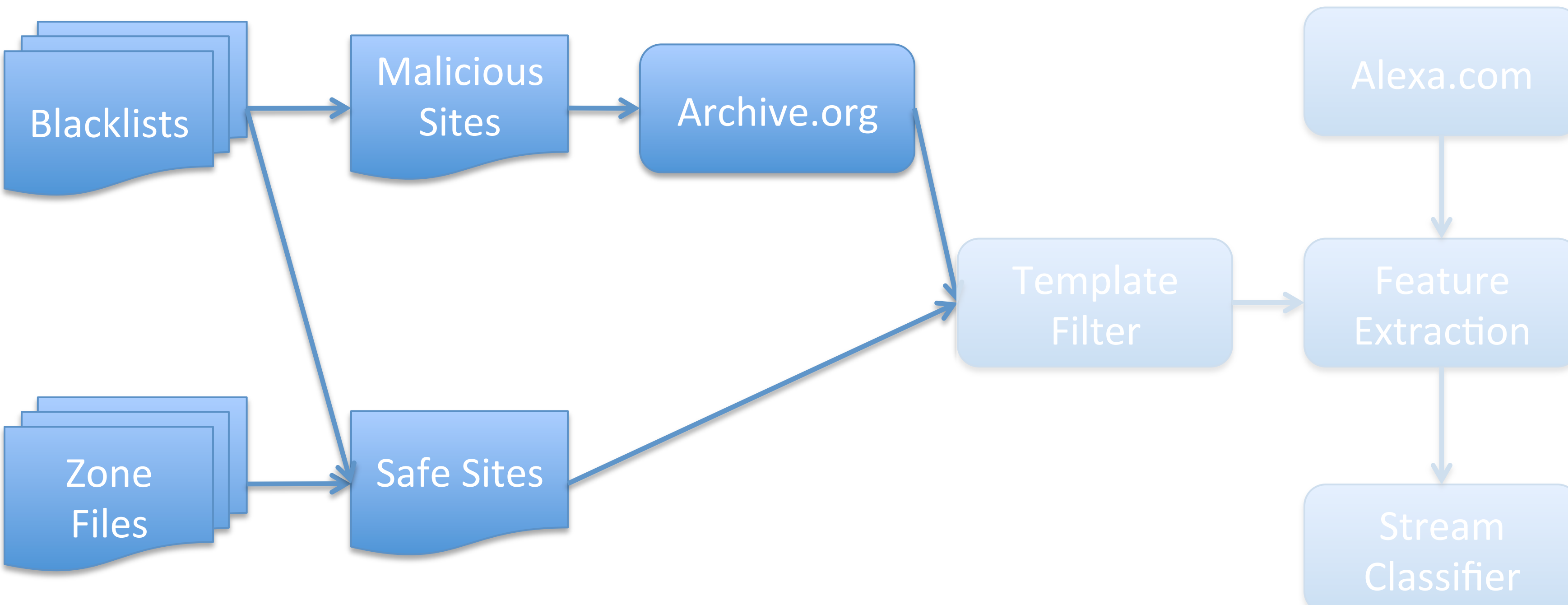
System Design



System Properties

- Efficiency
  - Internet dataset
- Interpretability
  - Need to build intuition about why the site will become compromised
- Robustness to Imbalanced Data
  - Far more benign examples than malicious ones
- Robustness To Mislabeled Data
  - Blacklists may contain errors or be incomplete
- Adaptive
  - Internet is a concept drifting, requires active adaptation

Dataset


Blacklists
Malicious Sites
Archive.org
Alexa.com
Zone Files
Safe Sites
Template Filter
Feature Extraction
Stream Classifier

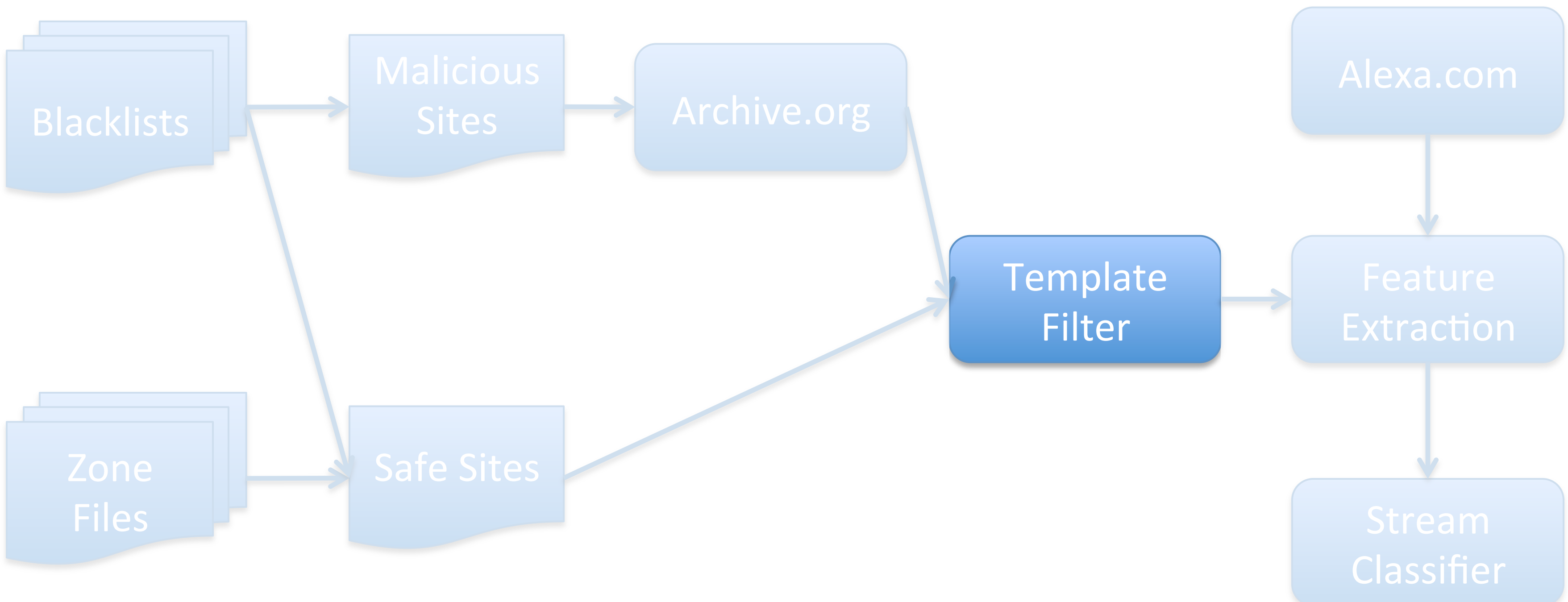
Dataset

Type	Instances	Archived Instances	% Archived
PhishTank	91,555	34,922	38.1
Search Redirection*	16,173	14,425	89.2
.com Zone Files	336,671	336,671	N/A

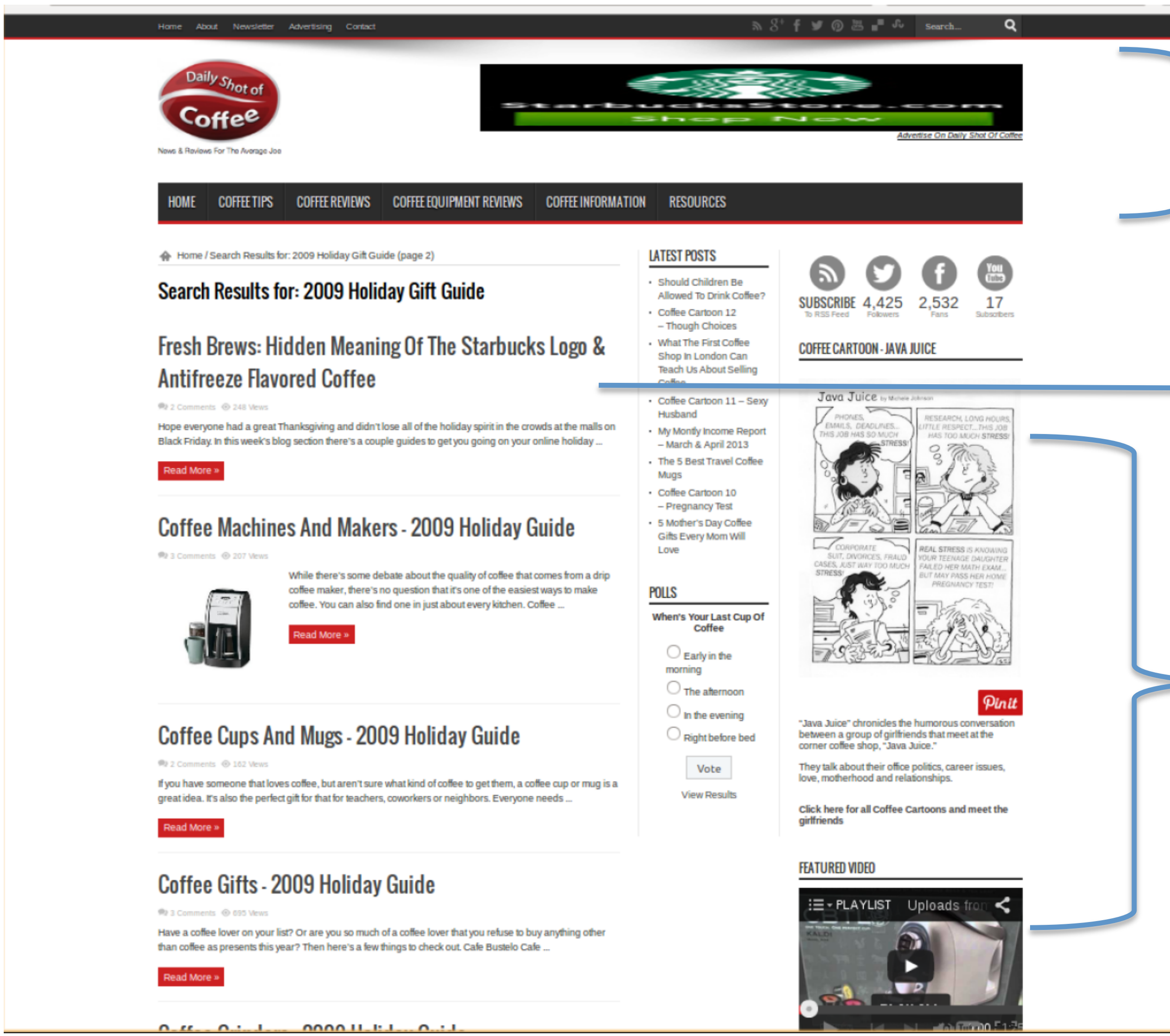
- PhishTank: Feb 2013 – Dec 2013
- Search Redirection: Oct 2011 – Sept 2013
- Zone Files: Feb 2010 – Sept 2013

* Leontiadis et al., 2014

Filtering


Blacklists
Malicious Sites
Archive.org
Alexa.com
Template Filter
Feature Extraction
Zone Files
Safe Sites
Stream Classifier

Filtering



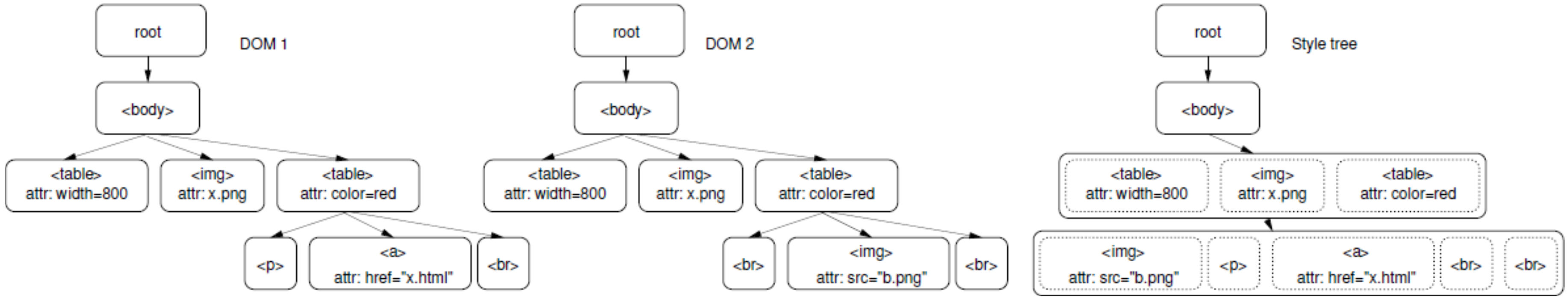
Navigation

User Content

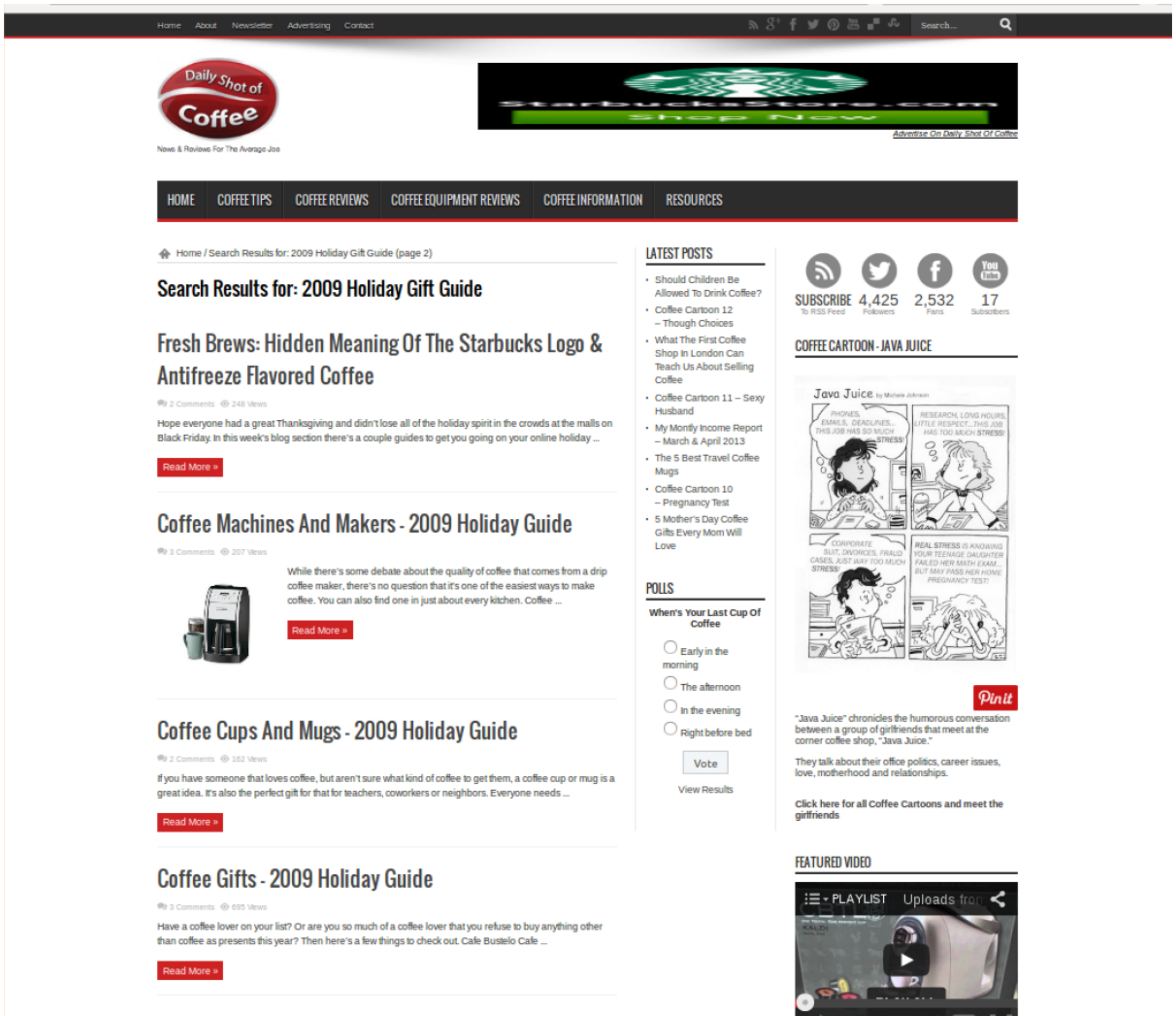
Social Media Links

Filtering – Based on [Yi et al., 2003]

- Compute entropy-like heuristic "Composite Importance" (CmpImp $\in [0, 1]$) for each element on a page
- Remove elements above a fixed threshold

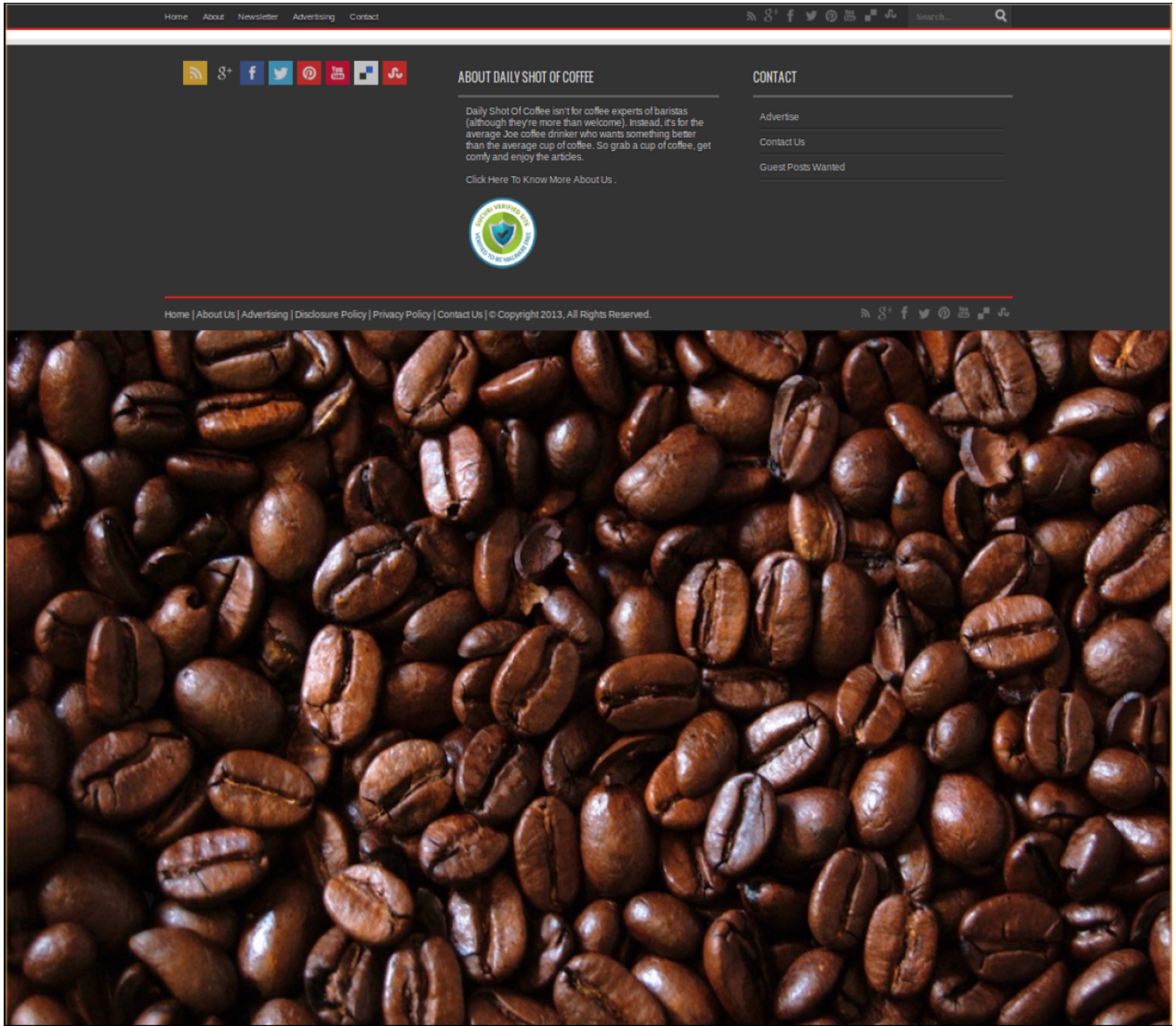


Original Page

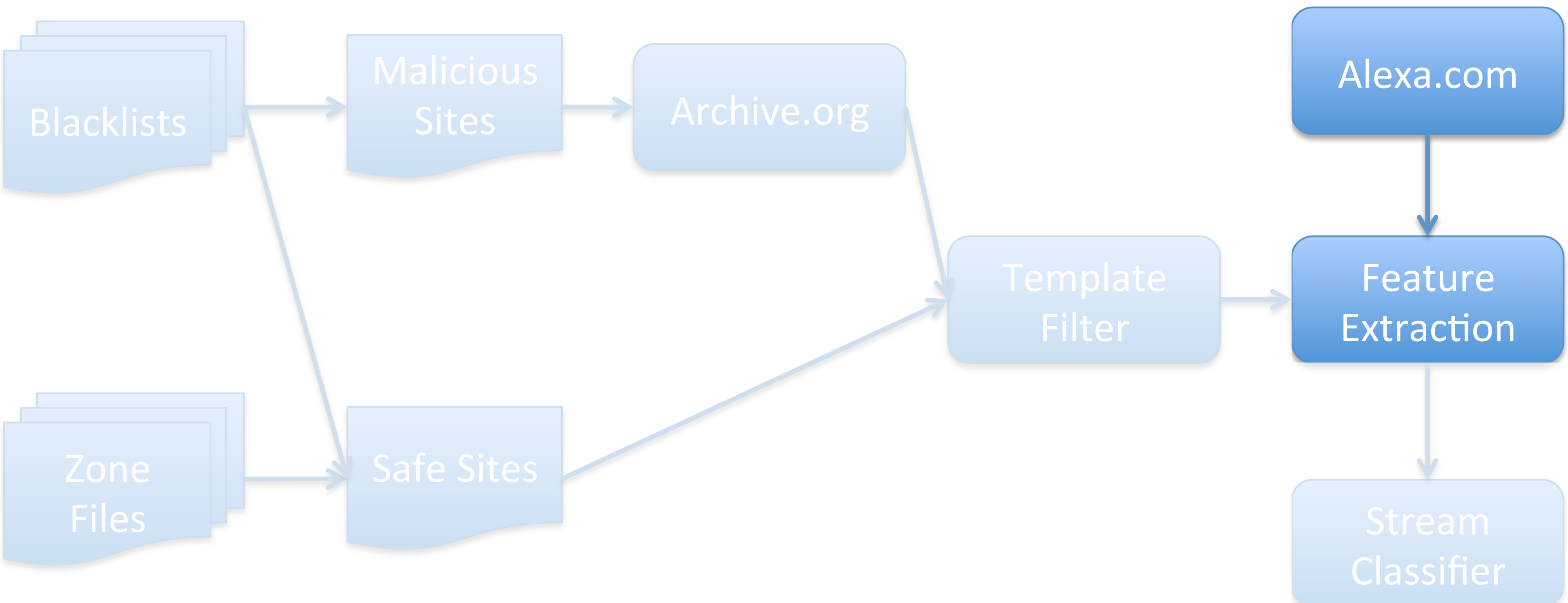


Threshold = 0.99

Threshold = 0.1



Feature Extraction



Feature Set

- Traffic Features
  - Site Rank
  - Links into site
  - Load Percentile
  - …more
- Content Features
  - HTML Tags (type, content, attributes)

Dynamic Features

- Millions of unique HTML tags (including content)

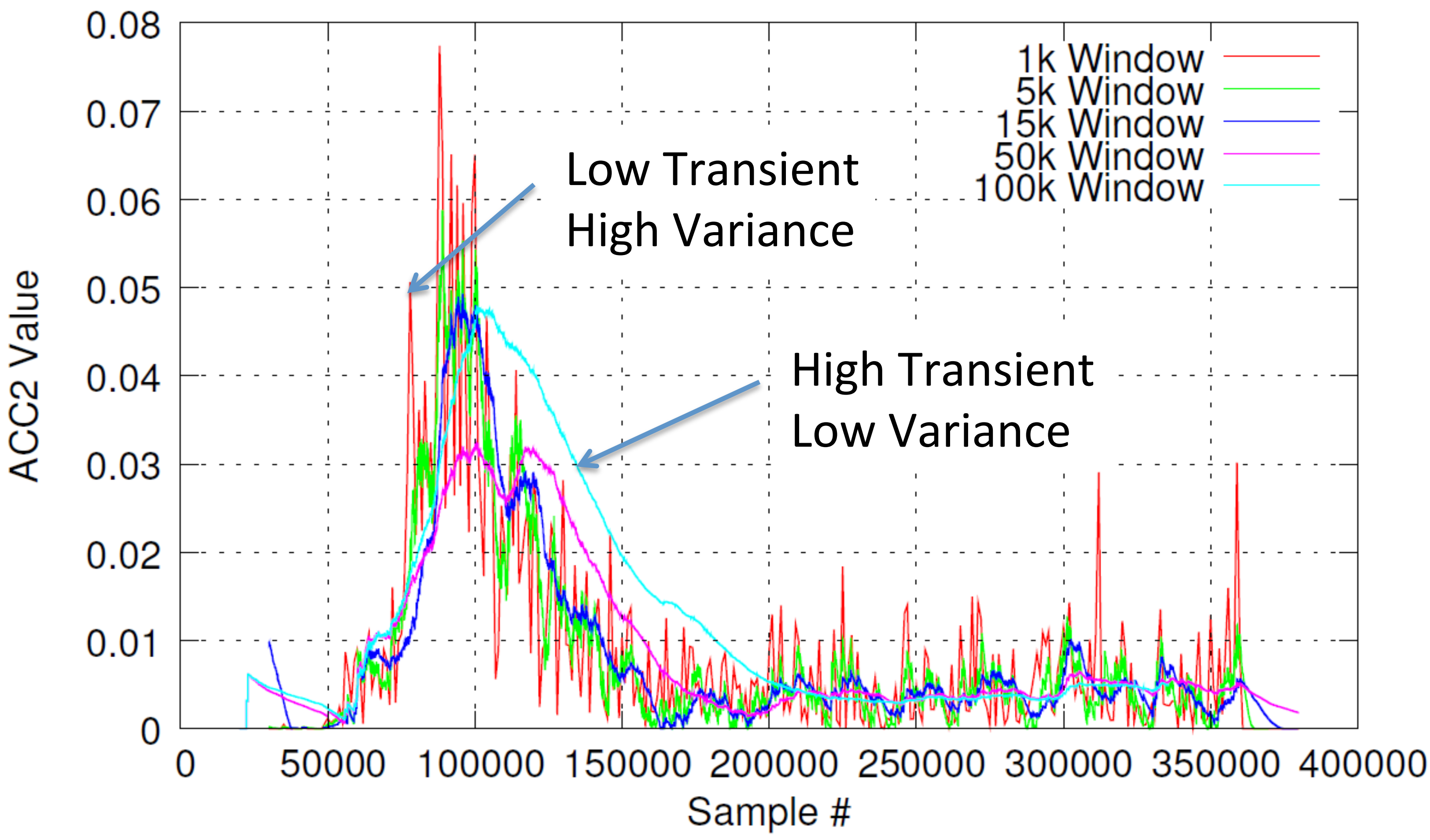
- Solution: order tags by some statistic, select top N
  - ACC2 based on [Foreman, 2003]
  - Let $\mathcal{B}$, $\mathcal{M}$ denote the set of benign and malicious sites respectively, w the set of tags from a site, then ACC2 for a tag x can be defined as:

$$s(x) = \left| |x \in w : w \in \mathcal{M}| / |\mathcal{M}| - |x \in w : w \in \mathcal{B}| / |\mathcal{B}| \right|$$

Prominent Features After 90,000 Samples

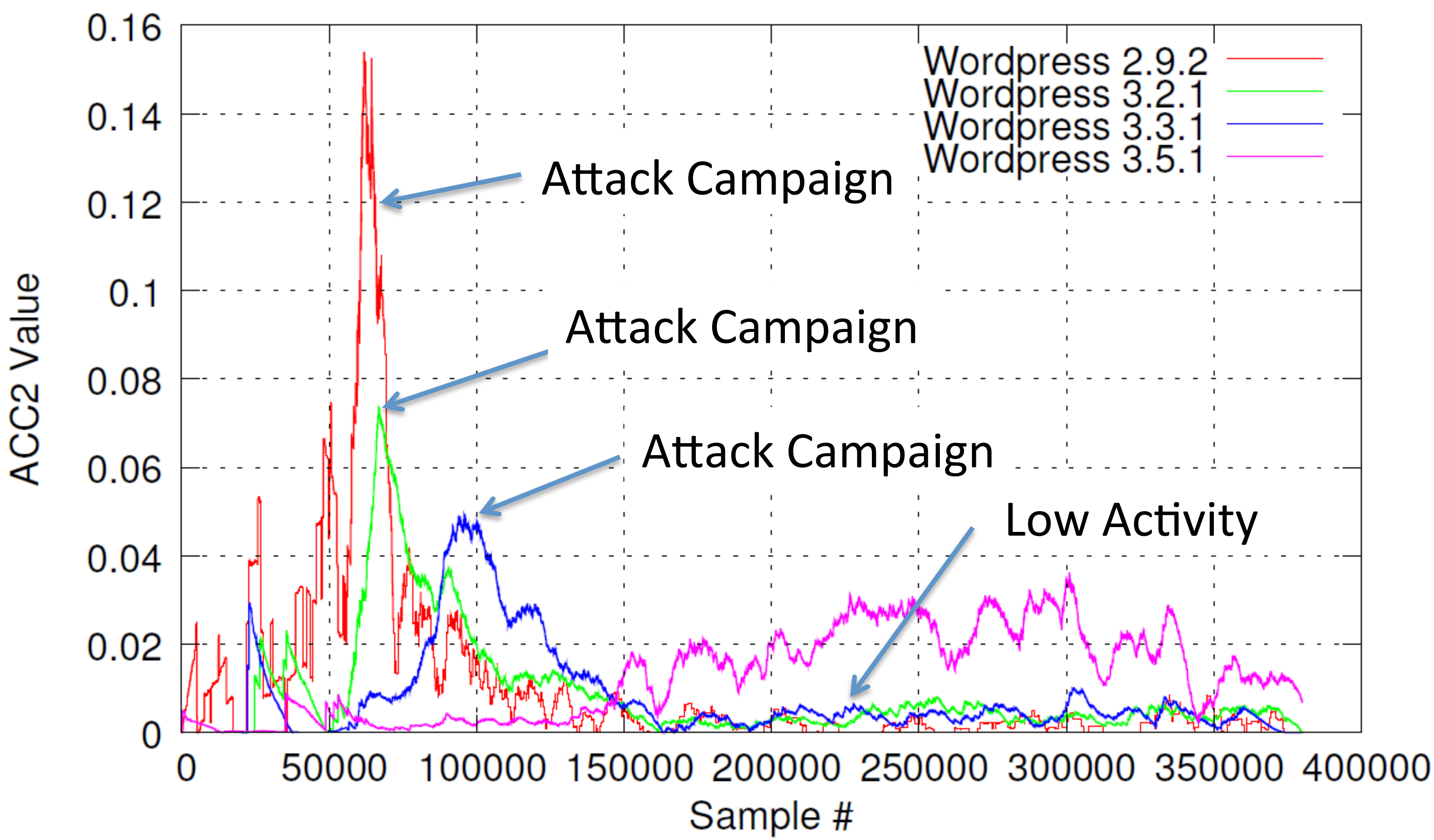
Feature	Statistic Value
meta{‘content’: ‘Wordpress 3.2.1’, ‘name’: ‘generator’}	0.0569
ul{‘class’: [‘xoxo’, ‘blogroll’]}	0.0446
You can start editing here.	0.0421
meta{‘content’: ‘Wordpress 3.3.1’, ‘name’: ‘generator’}	0.0268
/all in one seo pack	0.0252
span{‘class’: [‘breadcrumbs’, ‘pathway’]}	0.0226
If Comments are open, but there are no comments.	0.0222
div{‘id’: ‘content_disclaimer’}	0.0039

Varying Window Sizes

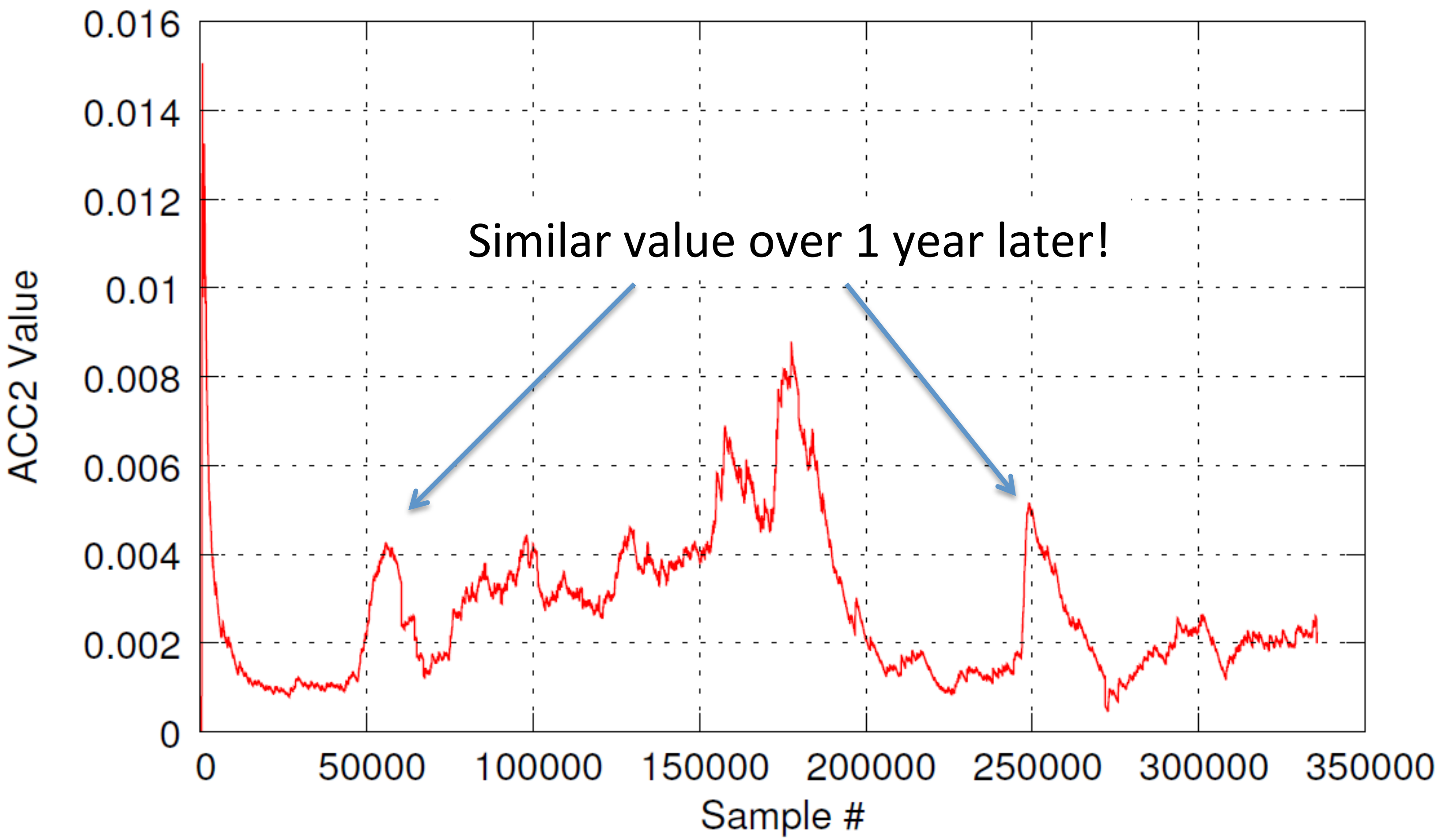


Feature: meta{'content': 'Wordpress 3.2.1', 'name': 'generator'}

CMS Evolution

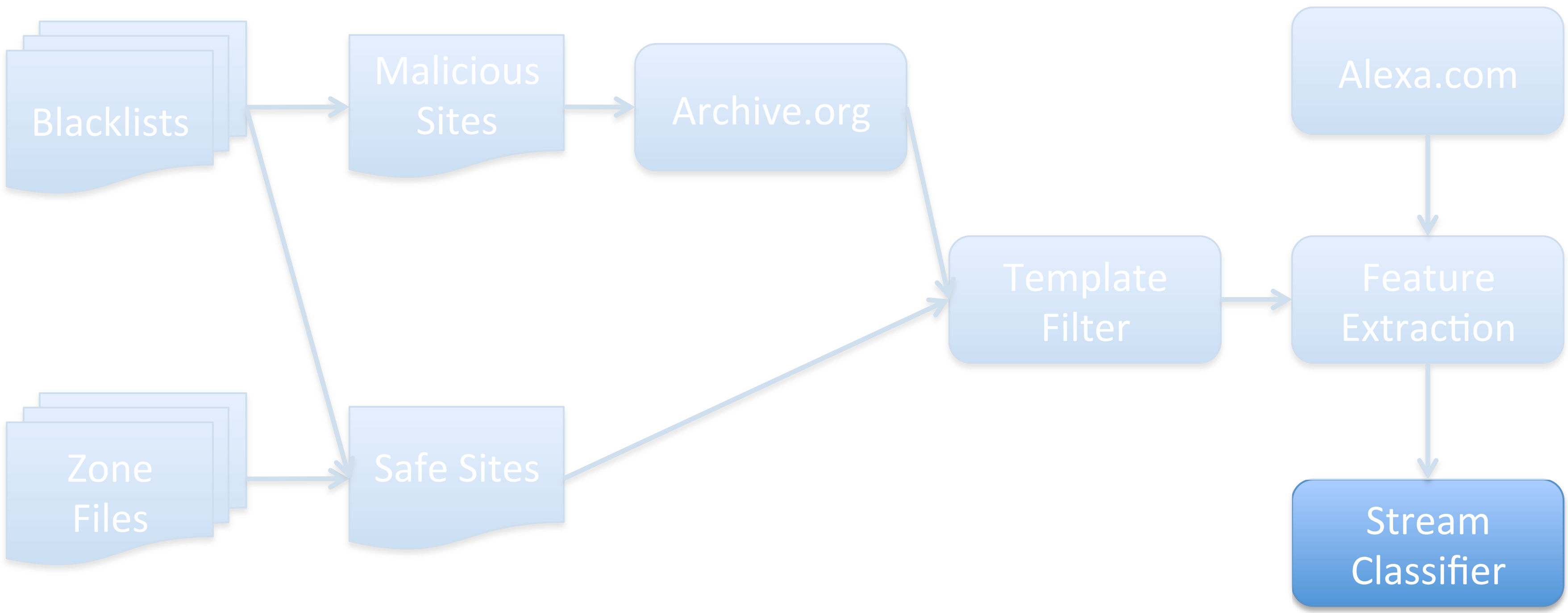


Parking Page Feature



Feature: div{'id': 'content_disclaimer'}

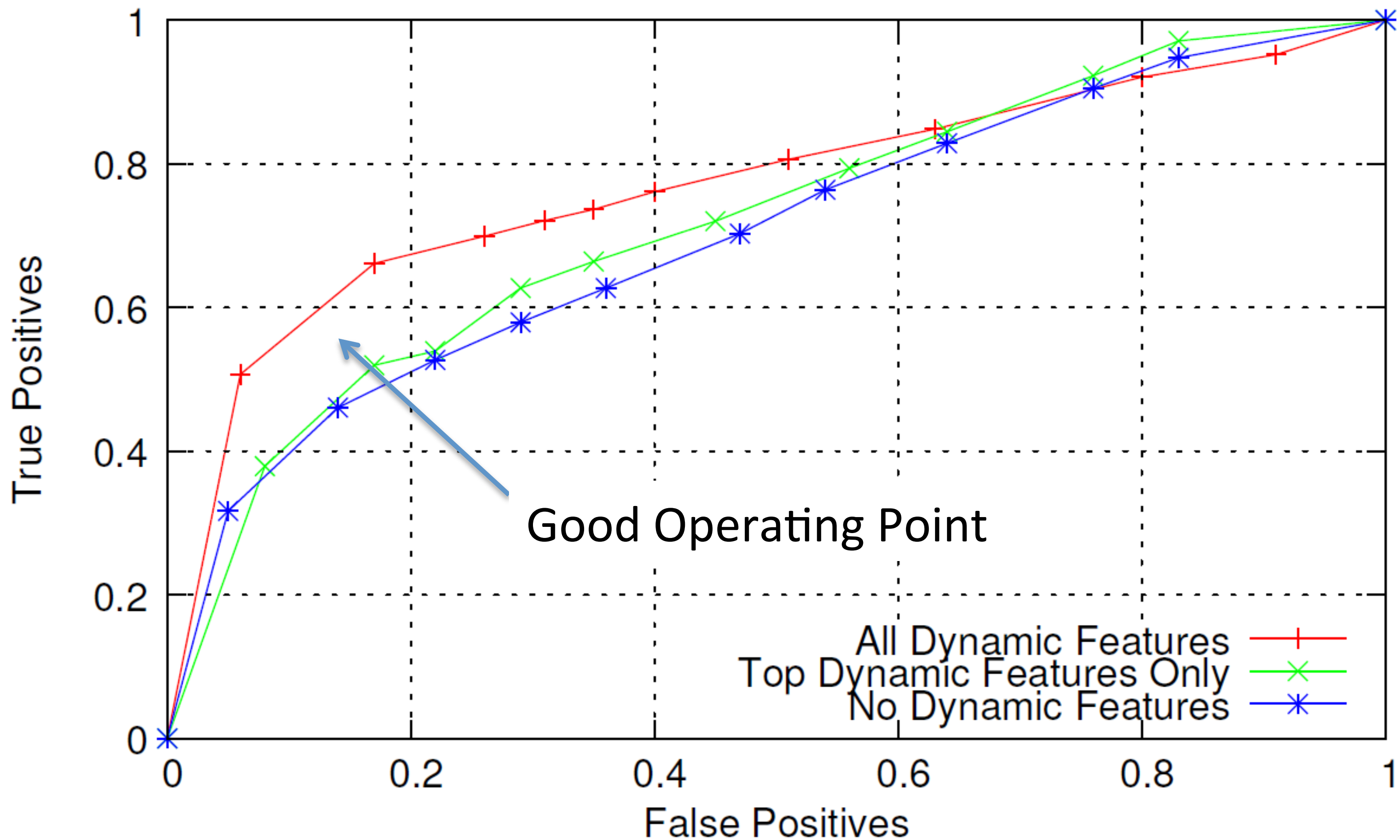
Classification



Classification

- Largely based on [Gao et al., 2007]
- Break input data stream into blocks
- Resample input blocks
- Train ensemble C4.5 decision tree classifiers using Hoeffding bounds [Domingos et al., 2000]
- Retrain periodically using new dynamic features

Classification Results



Limitations

- Only makes sense when page content and traffic statistics are risk factors of malice
  - Sites hacked via weak passwords or via social engineering attacks violate this
  - Sites that are maliciously hosted may violate this

- Requires some sites to become compromised in order to make predictions

Conclusions

- Predicting websites that become malicious in the future is possible!
- Acceptable performance can be achieved even on our modest dataset

Kyle Soska – ksoska@cmu.edu