



Motivation

- Trackers rely on cookie syncing to share their view of a user's browsing history with other trackers
- Cookie syncing circumvents the Same-Origin Policy to share one tracker's identifier cookie with another tracker
- After cookie syncing, trackers conduct back-end data join using the linked identifier cookies
- The Problem:** Cookie syncing is nontrivial to detect. Requires knowledge of known identifiers to confirm positives instances, but identifiers can be hashed or otherwise obfuscated.

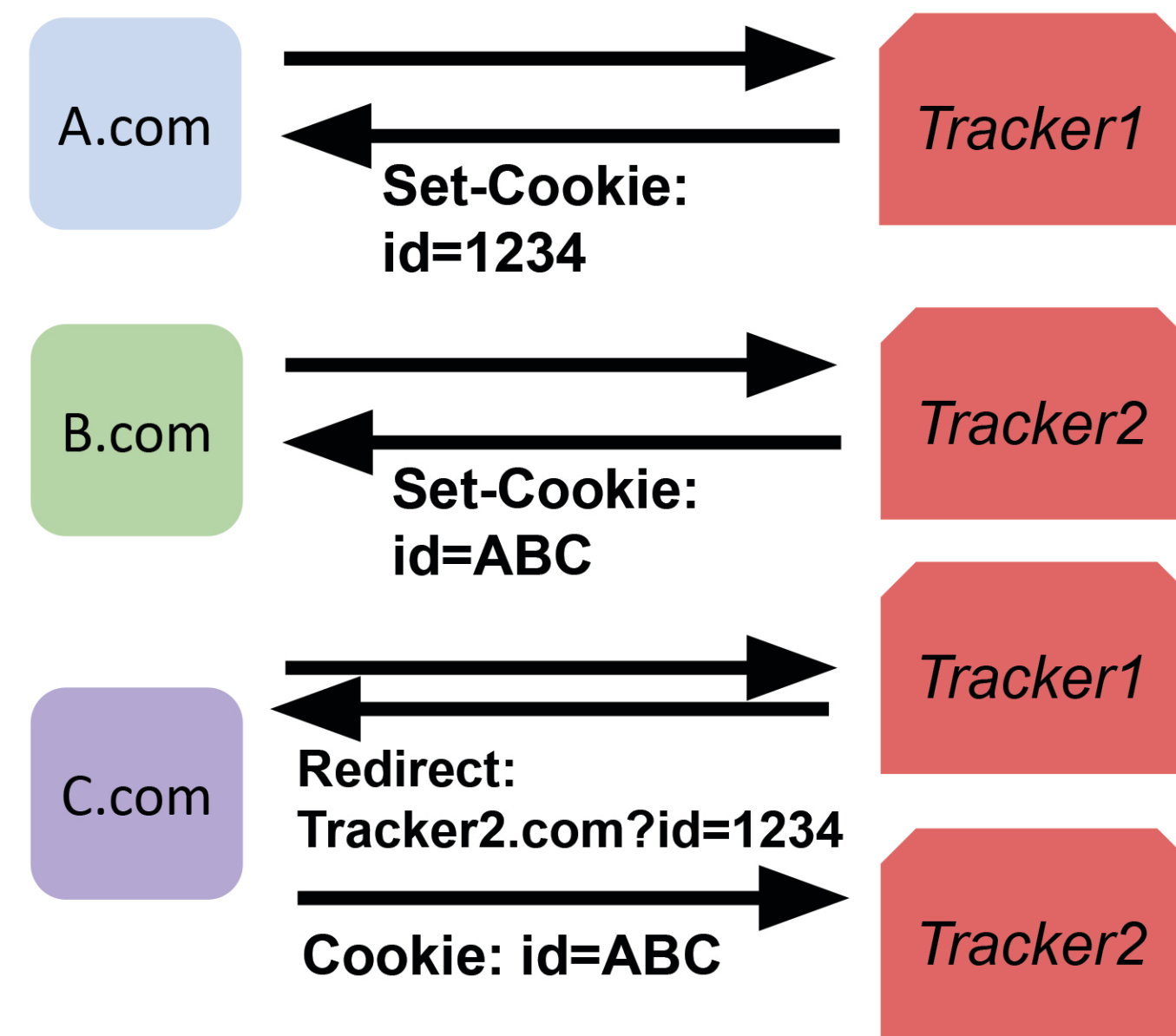


Figure 1: How Cookie Syncing Works

Our Contribution

- Prior work has relied on known tracker domain names to detect cookie syncing endpoints [1], which is not suited to detect novel cookie syncing endpoints.
- Our main idea is to leverage cookie syncing endpoint's distinguishing characteristics in the HTTP *redirect graph*
 - Redirect graph: directed graph of redirected domains (nodes) connected by HTTP requests (edges)
- We propose a graph based ML model to detect cookie syncing endpoints *without needing to rely on identifiers or prior knowledge of known tracking endpoints*

Classification Task

- Input Graph Features:** degree centrality, in-degree centrality, out-degree centrality, PageRank, etc.
- Output Classes:** Cookie Syncing vs. Not Cookie Syncing

State-of-the-Art

- Papadopoulos et. al. [1]: decision tree classifier of cookie syncing. Does not consider the presence of identifiers as a feature
 - Classification goal: **endpoints** vs. requests
 - Dataset: **26K HTTP redirects** vs. 825.6K HTTP requests
 - Feature selection (feature number): **graph-based (4)**, HTTP request based (4), *domain name* (1)
 - Most informative feature: **PageRank** vs. domain name
 - Target Classes: +/- **cookie syncing** vs. +/-/else cookie syncing

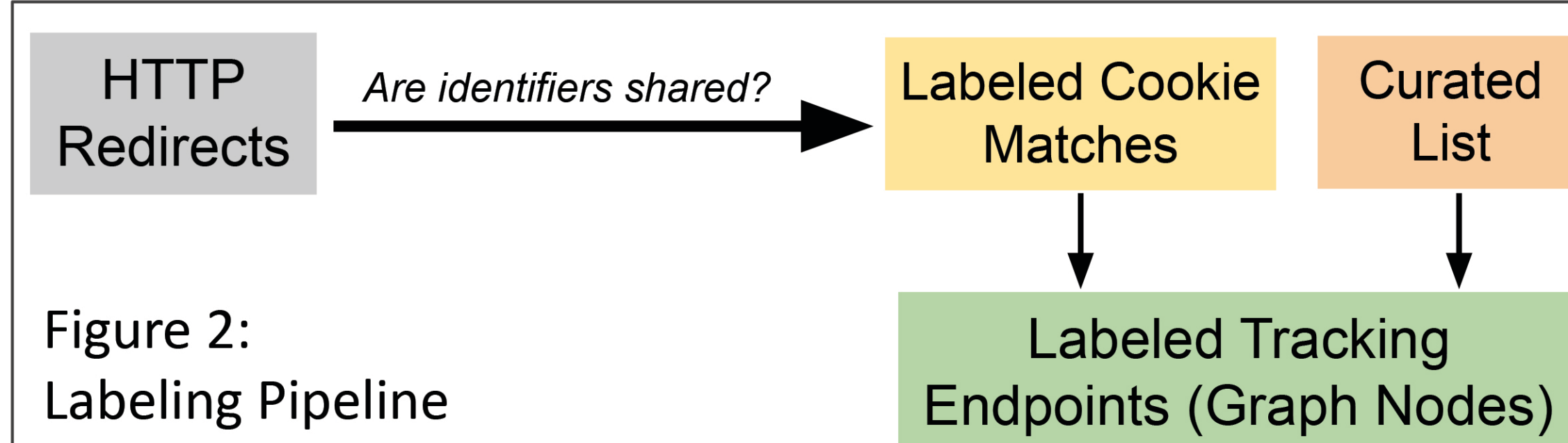


Figure 2: Labeling Pipeline

Dataset

- OpenWPM crawl of Alexa top 1K sites + 9K random sites from top 100k
- Ground Truth Labeling:**
 - Request labeling:** *requires the sharing of a known user identifier* to be labeled as positive for cooking syncing
 - Propagating request labels to endpoint labels:** determined by consolidation of each endpoint's labels + manually curated list of known cookie syncing endpoints
- Dataset Statistics:**
 - 26K labeled nodes (4% positive, 25% negative, 71% unknown)

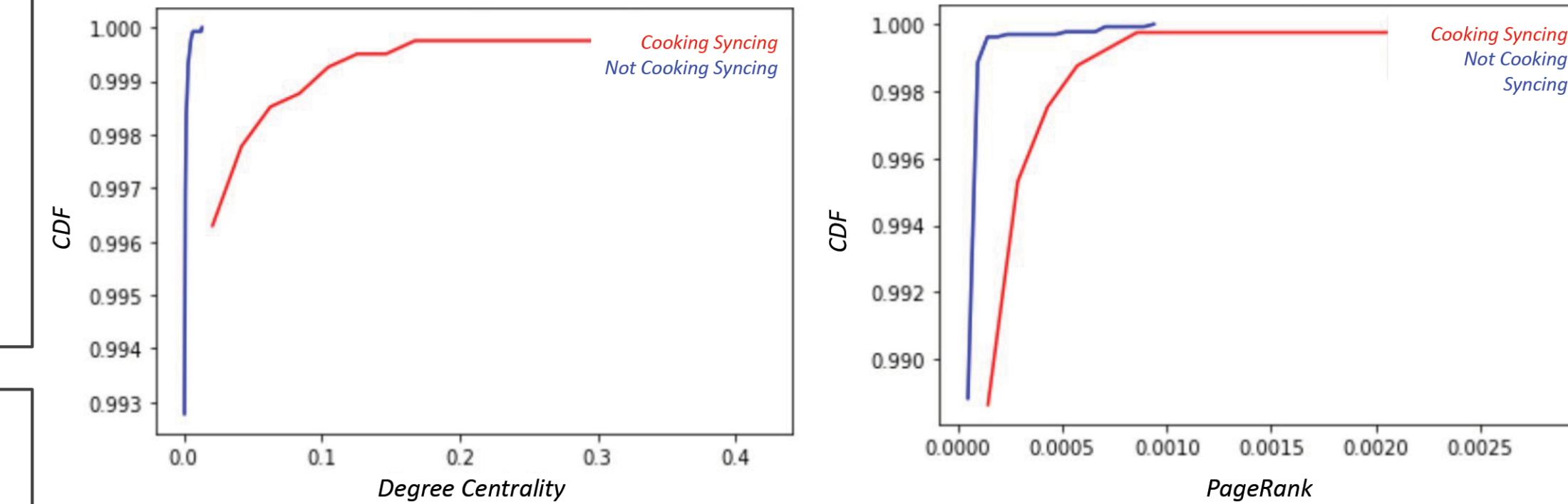


Figure 3

Results

- Papadopoulos [1] classification model
- Our graph-based features + Random Forest' model
- Takeaway: Results indicate the potential of graph-based features to detect cookie syncing endpoints

Model	AUC	PR	RC	F1	CV Test Score
Pap. [1]	.887	.845	.840	.834	NA
Our	.890	.855	.503	.632	.918

Figure 4: Model Evaluation Using 10-fold cross validation

Future Work

- Next Steps**
 - Evaluate on larger sample size
 - Our model outperforms previous Precision and AUC (TP/FP) scores; fails to meet Recall (TP/TP+FN) and F1 benchmarks.
 - Hypothesis: Recall and F1 scores are lower due to sample size difference between the positive and negative labels.
 - Comparison with Papadopoulos [1] on unlabeled domains
 - Design new graph based features
 - Try deep learning based graph classification models

