Graphing Crumbling Cookies

Matthew Malloy
Comscore
mmalloy@comscore.com

Jon Koller
Comscore
jkoller@comscore.com

Aaron Cahn
Comscore
acahn@comscore.com

ABSTRACT

Device graphs are datasets that organize and associate the many identifiers produced by PCs, phones, TVs, and tablets as they access media on the internet. Digital *cross-media*, the delivery and measurement of advertisements across screens, has grown increasingly reliant on device graphs. In response to privacy and tracking concerns, some web browsers limit the persistence of the identifiers used in device graphing. Examples include Safari's implementation of *Intelligent Tracking Prevention* and user invoked *incognito/private* browsing capabilities. Non-persistent identifiers create both a scale and accuracy challenge for device graphing. Motivated by accurate *audience measurement*, this paper demonstrates how measurement and other entities in the digital advertising ecosystem can overcome the lack of persistence of identifiers without the need for techniques such as browser fingerprinting. The approach is based on first constructing a device graph and applying community detection using persistent identifiers, and then appending non-persistent identifiers to the original communities using a technique termed *graph backfilling*. The resulting device graphs are of immense scale, organizing more than 4.7 billion identifiers worldwide.

CCS CONCEPTS

• **General and reference** → *Measurement*; • **Information systems** → *Online advertising*;

KEYWORDS

Internet measurement, device graphs, online advertising

ACM Reference format:
Matthew Malloy, Jon Koller, and Aaron Cahn. 2018. Graphing Crumbling Cookies. In *Proceedings of AdKDD The 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, Anchorage, Alaska, USA, August 19–23, 2019 (AdKDD '19),* 6 pages.
https://doi.org/10.1145/3219819.3219852

1 INTRODUCTION

Cross-media audience measurement refers to the measurement of media and advertisement consumption *across* screens, including PCs, phones, tablets, and smart TVs. The delivery of digital media and advertisements, while platform dependent, involves a complex interaction between a client device (i.e., PC, mobile phone or smart TV) and publishers, advertising networks, advertising exchanges, advertisers, and audience measurement companies. These entities collect information about devices as they request media and interact with the advertising ecosystem. The information is used for a variety of purposes including audience measurement, attribution, auditing, fraud detection, and targeted advertising. The information is also used to address the primary challenge of cross-media – identification of a unique person across screens.

Absent login information, third party measurement entities rely on two types of identifiers to track activity: *i)* third party browser cookies, and *ii)* operating system (OS) level mobile advertising identifiers (*e.g.,* Apple's Identifier For Advertisers (IDFA) and Android's Advertising ID). Third party cookies, or *cookie IDs*, can identify a user when they accesses the web from a web browser. OS level advertising IDs, or *ad IDs*, facilitate identification of a user when mobile apps and smart TVs access resources on the internet. We refer to both types of identifiers generically as *IDs*.

While both types of IDs facilitate measurement and targeting in the advertising ecosystem, they differ drastically in the practicalities of their use. Cookie IDs are domain specific. Consider the following hypothetical example: a user visits *shoes.com*. The user then visits *trucks.com*. The browser itself ensures a tenant of web security: *shoes.com* and *trucks.com* have different cookie IDs for the same device/browser. This makes it challenging for parties to identify users that are interested in shoes *and* trucks without further effort in relating the different IDs. This is largely overcome by deploying third party tracking and measurement *tags*. For example, both *trucks.com* and *shoes.com* may direct a browser to *measurement.com* in the background, which has the same *third party* cookie ID when re-directed from either domain. In practice, some domains call tens of tracking and measurement domains in the background [9], many of which re-direct to other measurement domains, in a practice called *cookie syncing*.

Ad IDs are accessible by apps. Any app installed on a device, provided a limited set of restrictions are met, can access the OS level ad ID. Unlike cookies and domains, different apps will observe the same ad ID for the same device. Hence, app developers can share information tied to ad IDs with relative ease. Consider the following example: the developers of a shopping app sell purchase information connected to an ad ID to a third party. The developers of a travel app do the same. The third party is immediately able to find common IDs that are interested in travel and shopping, and make that information available for others to use. A cottage industry has grown around purchasing and selling information tied to ad IDs. Both Google and Apple developer agreements have terms that restrict usage; it remains unclear how enforcement is possible.

Tracking and user identification on the web has been given significant attention by internet privacy advocates. As the primary mechanism for cross domain tracking, the third party web cookie

AdKDD '19, August 19–23, 2019, Anchorage, Alaska, USA

ACM ISBN 978-1-4503-5552-0/18/08…$15.00
https://doi.org/10.1145/3219819.3219852

has acted as a lightning rod to internet privacy advocacy [27, 35]. In response, browser developers have created mechanisms to reduce the efficacy of the third party cookie. The release of iOS 11 includes "Intelligent Tracking Prevention" [3], which aims to reduce tracking of users by third parties by frequent resetting of cookie IDs. The result, from the perspective of a third party measurement entity, is an increasing number of increasingly *ephemeral* IDs. Browser features such as 'incognito browsing' in Google's Chrome browser and 'Private Browsing' in Safari also create *ephemeral* cookies that only exist for the duration a browsing session. An *ephemeral* ID lasts, approximately, for a time period less than a day.

Internet device graphs are datasets that capture relationships between IDs at scale. By associating IDs that belong to the same household, user, and device, device graphs can be used to solve the fundamental challenge of cross media measurement – identifying users across screens. Previously published work on internet device graphs [16, 22] has not addressed the issue of capturing ephemeral identifiers. Instead, previous work has explicitly excluded these identifiers as they result in a scale that is not amenable to graphing and community detection.

Motivated by cross media audience measurement, this paper addresses the scale related challenges caused by including ephemeral IDs in device graphing processes. Our approach, termed *backfilling*, expands on the work in [22] and [16] and allows the number of IDs organized by the graphing process to grow from 1.7 billion nodes to 4.7 billion nodes (across twelve countries). The approach first separates observed IDs into two sets *1)* persistent ID and *2)* non-persistent or ephemeral IDs. The persistent IDs are graphed as described in [22], resulting in household level cohorts. Any IDs that are not assigned a cohort, including non-persistent IDs, are eligible to be associated with the cohorts by 'backfilling'. The backfilling process, in short, associates cohorts with a hashed IP addresses at relatively short times scales. The IDs associated with the hashed IP are then assigned to the cohort, provided certain conditions of the algorithm are met. It is notable that the approach scales with the number of nodes in the graph, not the number of edges.

We showcase our methodology with a large dataset of more than 1.8 trillion internet events collected over the course of six weeks from Comscore's *digital census network*. Comscore's census network is comprised of web page, mobile application, advertisement, and video tags deployed across the internet. The tags generate server logs of client requests for web pages, actions in mobile applications, video requests, and advertisement deliveries. The dataset includes 3.4 billion persistent IDs, and more than 7.8 billion non-persistent IDs across twelve countries. Of these 11 billion IDs, the original graphing process organizes 1.7 billion IDs into small cohorts that approximate residential households, and then appends 3 billion additional IDs to those cohorts though backfilling. The system is deployed in a production Hadoop Map Reduce environment, and runs weekly. We report on results and the details of the implementation in Section 3.

Finally, given the rapidly changing digital advertising ecosystem, we discuss privacy and ethics associated with device graphing in depth in section 4. Our use case is accurate *audience measurement* and we advocate that privacy considerations are different for different applications. Our aim is to shed light onto difficult privacy questions and provide context for ongoing conversation. We also extrapolate on three important privacy related aspects of our approach. First, devices that return a blank or null ID are excluded in the process. Second, we do not attempt to fingerprint devices. Third, our device graphing approach is termed *probabilistic* (see [22]) meaning the result is incorrect a non-negligible percentage of time, providing plausible deniablity.

To summarize, this paper makes a number of contributions. First, we propose an algorithm for device graphing that scales proportionally to the number of nodes in the graph, not the number of edges. To the best of our knowledge, this is the first study of the problem of grouping multiple IDs at a scale exceeding four billion identifiers. We demonstrate the efficacy of our approach on unique datasets of immense scale, report on basic characteristics of the output, and validate using a unique ground truth dataset.

2 METHODOLOGY

This section formalizes notation and defines the problem. $G = (V, E)$ is a graph, with nodes V and weighted edges E. $e \in E$ is a weighted edge, which is a two element subset of V with a real valued weight: $e = (i, j, w) \in V \times V \times \mathbb{R}$. A node $i \in V$ is a an ID (*e.g.,* a web cookie or advertising ID). Groupings of nodes are termed cohorts $C_j = \{i, \dots\} \subset V, j = 1, \dots,$ and satisfy $C_m \cap C_n = \{\}$ for any $m \neq n$. In this paper, we associate additional IDs with the cohorts. The cohorts, after additional IDs are included, are denoted C_j^+, where $C_j \subset C_j^+$. The nodes of the graph are partitioned into two sets: V_p, the set of persistent IDs, and V_np, the set of non-persistent IDs, with $V_\text{np} \cup V_\text{p} = V$ and $V_\text{np} \cap V_\text{p} = \{\}$. For our purposes, the approximate size of V_p is on the order of one billion nodes. V_np is significantly larger, approximately ten times the size of V_p.

Cohorts are small groups of closely related IDs. In this paper, the cohorts approximate groups of IDs that share a residential household or small business place. The cohorts can be tuned to approximate person level groups or larger internet communities.

Device graphing faces significant scale challenges: the size of the graph can surpass 10 billion nodes, beyond the scale at which even the simplest community detection algorithms can run efficiently. Previous efforts in device graphing have dealt with this issue by dividing the node set V into V_p and V_np and then simply discarding V_np [22]. This paper addresses that shortcoming by answering the following question: how can $i \in V_\text{np}$ be associated with the cohorts $C_1, \dots,$ in light of the prohibitive scale of V_np? In other words, given a set of disjoint cohorts $C_1, \dots$ and a set of unassigned IDs $i \notin \{\cup_j C_j\}$, how can we use additional information to associate the unassigned IDs, $i \notin \{\cup_j C_j\}$ with the cohorts $C_1, \dots$?

2.1 Graphing

Starting with a corpus of data consisting of tuples of *(ID, IP-address, timestamp)*, V is defined as the set of IDs appearing in the corpus. V is partitioned into V_np and V_p using the time-stamps associated with each ID; for example, V_np is the set of IDs that have difference between the last and first timestamp less than twenty-four hours.

Next, the corpus of data is restricted to tuples *(ID, IP-address, timestamp)* associated with IDs in V_p. The IP co-location algorithm of [22] proceeds as follows. On the first epoch, for each IP in the dataset, an edge is created between every pair of IDs that share an

IP. The weight of the edge is the inverse of the number of IDs (in V_p) observed on the IP address. The process is repeated for T epochs, resulting in T graphs. The full graph, G, is created by summing the weighted edges across the epochs. Full details are presented in Algorithm 1 of [22].

Prior to community detection, low weight edges are discarded from the graph, resulting in the culled graph, denoted G_γ. The culled graph is defined as $G_\gamma = (V_\gamma, E_\gamma)$, with $E_\gamma = \{e_i : w_i > \gamma\}$. Any node with all edges below γ are excluded from the node set; $V_\gamma = \{i : \bigcup_j w_{i,j} > \gamma\}$. The parameter γ is used to tune and adjust the size of the output communities. The nodes in V_γ are termed *graphed* IDs, while IDs in $V \setminus V_\gamma$ are termed *orphaned* IDs.

After graph creation is complete, community detection algorithms are applied to G_γ resulting in cohorts that associate IDs from the same household, machine, or person. While [22] proposes use of Louvain Modularity [7], a number of community detection algorithms are appropriate. The result is a set of communities $C_1, \ldots$ with $\{\bigcup_j C_j\} = V_\gamma \subset V_\text{p}$. This process is summarized on the left side of Fig. 1.

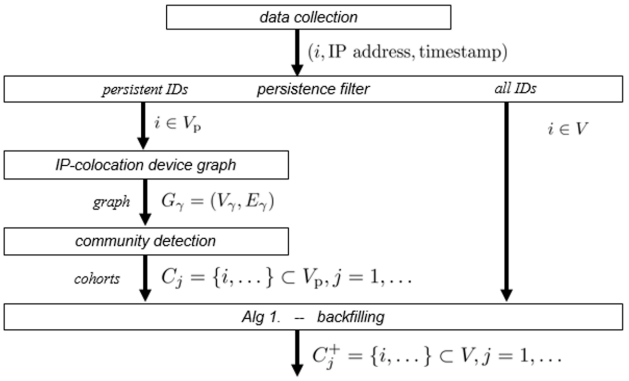

Figure 1: Device graphing with backfilling. The left side of the digagram shows the process described in [22] in relation to the backfilling algorithm described in Alg. 1.

2.2 Backfilling

The backfilling algorithm requires two parameters: α the persistent ID threshold, and γ, the edge weight threshold. The input to the algorithm is *i)* tuples of *(ID, IP-address, timestamp)* for all IDs in V, and *ii)* cohorts $C_1, \ldots$ that partition V_γ.

The backfilling algorithm begins as as follows. On epoch 1, each IP address that appears in the dataset is mapped to zero or one cohort C_j. The mapping is determined by the composition of *graphed* IDs that appear on the IP address. If graphed IDs on that IP from a single cohort C_j comprise a strict majority among all graphed IDs on that IP, then the IP is mapped to C_j. Next, all orphaned IDs (i.e, $i \in V \setminus V_\gamma$) on the mapped IP inherit the the same mapping. For each orphaned ID, a list of mappings to cohorts across IPs and epochs, denoted $[j^*_{i,k,t}]_{k,t}$ in Alg. 1, is maintained. The modal element of the list, if it exists, is chosen as the final assignment of the orphaned ID to a single cohort. The orphaned IDs assigned to C_j *and* the original elements of C_j together define C_j^+. The algorithm is described in Alg. 1.

Algorithm 1 Device Graph Backfilling

paramaters: minimum edge weight γ, persistence threshold α
input: observations: *(ID i, IP address k, time t)*
define V_p: set of persistent IDs, V_np: set of non-persistent IDs
build graph [22] $\{(i, k, t) \text{ for all } i \in V_\text{p}\}, \gamma \rightarrow G_\gamma = (V_\gamma, E_\gamma)$
community detection [22] $G_\gamma \rightarrow C_j, j = 1, \ldots$
for each IP k, each time step t
    define $I_{k,t} = \{i : i \text{ observed on IP } k \text{ on time } t\}$
    define $j^*_{k,t} = \{j : |C_j \cap I_{k,t}|/|I_{k,t} \cap V_\gamma| > 1/2\}$
    for each $i \in \{I_{k,t} \cap \{V \setminus V_\gamma\}\}$
        define $j^*_{i,k,t} = j^*_{k,t}$
$j^*_i = \text{mode}\,[j^*_{i,k,t}]_{k,t}$ for all i
$C_j^+ = C_j \cup \{i : j^*_i = j\}$ for all $j = 1, \ldots$

3 EVALUATION

In this section we apply Alg. 1 to an internet scale dataset provided by Comscore. The dataset is restricted to data collected in the US. We note that the numbers presented in this section are smaller than those presented in the introduction, reflecting the restriction to the US traffic. We begin with several paramaterizations of the persistent ID device graph of [22]. Our objectives are to demonstrate the merits and effectiveness of Alg. 1, and to validate the results against a unique ground truth dataset.

3.1 Data and Implementation

The data used to implement and validate our methodology is obtained from Comscore's digital network, one of the largest in the world. This data is collected through the deployment of either JavaScript/HTML tags or SDK tags across a wide variety of web pages, mobile applications, video requests, advertisement deliveries, and other distributed content. In the case of both techniques, a unique record is reported directly to Comscore's collection infrastructure when a client machine locally executes the tags.

For this study we use data collected over a 42 day (6 week) period. This data originates from the US between the dates of February 25, 2019 and April 7, 2019, with the epoch used being a UTC day. Alg. 1 was implemented on this data in Apache Pig [33], running on an Apache Hadoop environment with 500+ worker nodes. The parameters used to specify the initial persistent device graph were tuned. We report results for $\gamma = 0.4$ to 2.0 in 0.2 increments.

3.2 Characteristics of Backfilled Cohorts

The backfilled cohorts, for $\gamma = 0.8$, consist of 2.57 billion IDs, an increase of 1.74 billion IDs when compared to the original cohorts. This increase shows the effectiveness of the backfill process, and how it can be used to alleviate the challenges posed by ephemeral IDs. Table 1 shows values associated with Fig. 1 for $\gamma = 0.8$.

While the increase in graphed IDs demonstrates the effectiveness of Alg. 1, it also highlights the impact of γ, which in effect, creates a trade-off between the number of IDs assigned a cohort

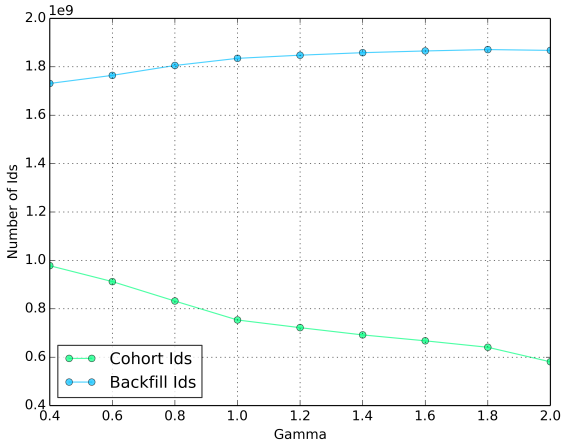

Figure 2: Number of IDs in initial cohorts and number of IDs added via backfill process in the US, plotted across various values of γ.

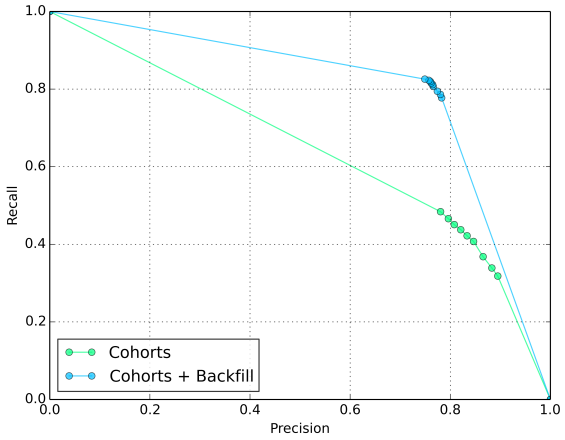

Figure 3: Mean precision-recall curve showing verification of cohorts, C_j and cohorts after backfill, C_j^+, against the ground truth communities in the US.

in the original graph processing, and the backfilling process. Increasing γ reduces the number of *graphed* IDs, but allows more IDs to be assigned during backfilling. Likewise, decreasing γ increases the number of IDs assigned a cohort in original graphing process, and decreases the number of IDs eligible for assignment during backfilling.

Fig. 2 shows the results of varying γ. Changing γ also has an impact on computational complexity of the process. As γ is decreased and the number of graphed IDs increases, community detection algorithms become prohibitive. At smaller γ, their implementation becomes all together intractable. Alg. 1 overcomes this problem; IDs that would have been discarded at higher γ are still eligible for inclusion in the cohorts.

Backfill Statistics, US	
raw rows	1.87T
all IDs , $\lvert V\rvert$	6.92B
persistent IDs, $\lvert V_{\text{np}}\rvert$	1.77B
graphed IDs, $\lvert G_{0.8}\rvert$	832M
number of cohorts, $\lvert\{C_1, \dots\}\rvert$	180M
ID count, initial cohorts, $\lvert\{i \in \cup_j C_j\}\rvert$	832M
ID count, cohorts via backfilling, $\lvert\{i \in \cup_j \{C_j^+ \setminus C_j\}\}\rvert$	1.81B

Table 1: Summary statistics of the device graphing and backfilling process in the US, $\gamma = 0.8$.

We note that of the total IDs in the US ($|V| = 6.92$ billion), 832 million are assigned a cohort in the original process, and 1.81 billion are assigned a cohort in the backfilling process. This leaves 4.28 billion IDs unassigned. Many of these IDs are excluded as they are not associated with an IP address assigned a mapping as defined in Alg. 1. The requirements of the association can be relaxed to allow for inclusion of more IDs.

3.3 Validation

Validation was achieved utilizing the Comscore Total Home Panel (THP), which collects data from participants' homes via customized wireless routers, providing a ground truth dataset. The customized routers capture statistics on web traffic for all device in the household, as well as both an obfuscated version of the media access control (MAC) address and any device IDs (3rd party cookies and advertising IDs) associated with the devices. Comscore's THP provides thousands of ground truth households, each consisting of one or more IDs. The IDs associated with a THP household are used as ground truth to to calculate *precision* and *recall* for the backfill process. Precision, defined as the fraction of IDs in a cohort that are also in the same ground truth household, and recall, the fraction of IDs in the the ground truth household that are also in the cohort, are defined formally in [22], eqn. (1) and eqn. (2).

Fig. 3 shows the precision-recall curve of the cohorts before backfilling, C_j, and after backfilling, C_j^+. The plot is generated by sweeping across selected values of γ. As expected, as original cohorts C_j are subsets of C_j^+, they exhibit significantly higher recall, while coming close to maintaining precision.

4 PRIVACY

Recent years have seen growing concerns over user privacy on the internet. While the concerns usually manifest as a discussion of cross domain tracking, privacy concerns related to digital IDs in general have been discussed on many forums, including academic literature, popular press, and government forums. Numerous articles have appeared calling for government regulation on tracking (*e.g.*, [17, 21, 31]). Legislation has been passed to address these concerns in various geographies, including the General Data Protection Regulation (GPDR) and the California Consumer Privacy Act (CCPA) of 2018. In the EU it is illegal to track users via IDs (either cookies or advertising IDs) without explicit consent [34]; this and other GDPR compliance requirements already have had an impact on third-party tracking [2]. In California, the CCPA of 2018

[6], which is slated to be enacted in 2020, while not as extensive as GDPR, requires companies that store IDs and associated information to enable opt out mechanisms that remove a user's IDs (i.e, cookies and advertising IDs) from the companies databases. A result of the legislation is companies voluntarily avoid doing business in these geographies prior to the enactment of new regulation.

Beyond government regulation, privacy concerns have sparked development of browser capabilities that limit the persistence of third party cookies. Starting with the release of iOS 11, Apple now includes *Intelligent Tracking Prevention* [3] (ITP) in the Safari web browser. ITP aims to reduce tracking by frequent resetting of third party cookie IDs. Tracking cookies are identified through a machine learning classification algorithm. Safari's implementation of ITP is irrelevant to many third party measurement domains since Safari rejects placement of any third party tracking cookie that is not associated with a domain directly visited by a user. This implies that most tracking entities (other than *google.com*, for example) are unable to even place a cookie. Apple has lauded it's own efforts to hinder third party tracking cookies through recent marketing campaigns [12]. The authors estimate that such efforts are symbolic at best, as no similar efforts have been taken by Apple related to the more persistent and universal IDFA.

While measures to increase privacy and reduce unwanted tracking on the internet are often seen as commendable, the issue is complex. As the driving financial engine for the internet, the advertising ecosytem has an obligation to capitalize on efficiencies. Such efficiencies include targeted advertisements and, our primary interest, accurate audience measurement. The ecosystem also has an obligation to respect privacy as specified by a user. While legislative measures and browser capabilities that limit tracking are seen by many as positive developments, this paper showcases that entities with sufficient data assets are able to reconstruct associations, making an argument that more can be done to limit tracking. The use case of such efforts play an important role in arguing for or against such techniques as acceptable. We posit that audience measurement statistics are an important, acceptable use case. The argument for targeted advertisements is a similar but distinct discussion.

The device graphing and graph backfilling processes described here exhibit a number of properties that conform with established privacy norms. First, the work herin, and in [16, 22], do not consider devices that return a blank or null ID, which is an indication that a user has selected to opt out of 'ad tracking' in either a browser or on a mobile device. In fact, these data points are actively excluded in the process. Second, the device graphing process we present does not rely on fingerprinting techniques, which often actively attempt to mitigate user privacy preferences. Also, it is worth noting that the device graphing approach here is *probabilistic* (see [22] for a detailed discussion). The validation results in prior sections imply the graph, while often correct, is indeed *incorrect* a non-negligible percentage of the time, providing an additional layer of privacy through plausible deniability. A number of other tools, including ad blocking software, help limit tracking and facilitate cookie management and removal (*e.g.,* [30], [23]). In addition to measures that limit tracking, numerous studies have focused on understanding the prevalence of tracking via IDs and the related issue of information leakage (*e.g.,* [9, 14, 19, 20, 23, 24, 26]).

Incomplete device graphs result in inefficiencies throughout the digital advertising ecosystem. Browser features that limit ID persistence further favor entities with access to first-party deterministic graphs based on email, login, or other information (for example, Google, Apple, Facebook and others). Third-party measurement companies, such as Comscore, face increasing challenges in providing accurate, independent tools for measurement of audience size and validation of delivery of digital advertisements. Some advertisers not privy to deterministic graphs estimate resulting losses into the hundreds of millions of dollars [5]. Regardless of perspective, device graphs will continue to play a role in online advertising. Making accurate probabilistic graphs in the face of ephemeral IDs is a challenging problem. We posit that by describing techniques for making these associations, we can expand the conversation about how best to describe and implement privacy policies to protect the privacy requests of users.

5 RELATED WORK

There is limited academic literature related to device graphing used for audience estimation beyond [16, 22]. Work in device graphing and more generally identification on the internet can be divided broadly into two categories: *i)* related work directly pertaining to device graphing, much of which is not published in academic literature, and *ii)* device fingerprinting techniques, which have a much more extensive history in academic publication.

As discussed in Section 2, our work is an extension of two papers in device graphing [16, 22]. The first of the two papers by Malloy *et al.* is the starting point for the algorithms described here. That work provides a generic framework for construction of a device graph based on colocation. The approach in [22] uses IP-colocation *i.e.,* the coexistence of two IDs on a given IP address, observed longitudinally in time, to establish relations between IDs, which was also summarized in Section 2.2. It then employs a community detection method for creating cohorts that approximate groups of IDs that belong to the same residential household. While this method is the basis for the backfilling approach described here, we only require that some grouping of persistent identifiers is done – cohorts that approximate households are not required. The second paper by Malloy *et al.* assumes that coarse associations have been made using colocation, and further refines the graph to create *user* and *device* level groupings. The approach relies on a supervised machine learning approach – in particular, an extension of Naive Bayes. Beyond these two publications, little academic or no literature exists with the taxonomy 'device graph'.

Commerical literature with the 'device graph' taxonimy is much more extensive, as commercial device graph offerings are widely available *e.g.,* [1, 4, 37]. Our work can be viewed as extension, improvement, and formalization of many of the ideas encapsulated in the commercial device graph literature. While there is limited academic literature using the taxonomy 'device graphs' beyond [16, 22], there is academic work under the taxonomy 'cross device tracking'. [36] aims to provide a method to detect when cross device tracking occurs, and its prevalence. A privacy focused analysis of cross device tracking is presented in [38].

The ability to identifiy a user/browser has long been persued, and led to the development of a variety of approaches beyond cookies and advertising IDs. One example, the Evercookie, aims to store information in a number of different client-accessible locations that cannot be easily cleared by a client [18]. Adobe and Microsoft have created similar functionality, which has since been discontinued for privacy and security reasons [11, 25]. Other examples of alterntative identifiers include *device fingerprinting* techniques. These techniques use features readily accessible for unique association at both the browser level [13, 28] and device level (also termed cross-browser identification) [8, 10]. Perhaps the most popular approach is *canvas fingerprinting*, in which the browser draws an object on the HTML canvas and hashes and encodes the result to create an identifier [15, 29]. Cross-browser identification, the identification of associated users across browsers on a single machine, has relied on either IP address-based [8] or hardware and OS features [10], and is known to have varying levels of accuracy [32].

6 SUMMARY AND FUTURE WORK

This paper extends the work of [16, 22] by proposing a technique termed backfilling, that allows inclusion of the many ephemeral IDs (*i.e.,* web cookies and advertising IDs) in the device graphing process. The approach operates as a bolt-on process to the previous device graphing techniques, and drastically increases the scale of the relationships identified between devices. The process operates by associating cohorts of *persistent* IDs with a hashed IP address, and then linking *non-persistent IDs* to those cohorts via hashed IP address associations. We demonstrate our technique at internet scale, and present details of the implementation. The resulting dataset organizes and relates more than 2.6 billion IDs in the US, and 4.7 billion IDs worldwide, into cohorts that correspond to residential households. To validate the approach, we show the process greatly improves the accuracy of the device graphing process, exceeding 75% mean precision and mean 80% recall when testing against a ground truth dataset. In total, processing of the graphing algorithms including the backfilling process, validation, and evaluation phases in the US complete in hours in a large Hadoop cluster. The device graphing pipeline remains very modular - the backfilling process can be bolted on to any number of core methods for building an initial device graph using persistent IDs.

REFERENCES

[1] 2017. Drawbridge. http://www.drawbridge.com/c/graph/. (October 2017).

[2] 2017. The EU General Data protection Regulation. (October 2017). http://www.eugdpr.org/

[3] 2017. Intelligent Tracking Prevention. https://webkit.org/blog/7675/intelligent-tracking-prevention/. (October 2017).

[4] 2017. Lotame Cross Device. http://www.lotamecrossdevice.com. (October 2017).

[5] 2018. Apple Tracking Block Costs Advertising Companies Millions. *The Guardian* (January 2018).

[6] 2018. TITLE 1.81.5. California Consumer Privacy Act of 2018. (June 2018). https://leginfo.legislature.ca.gov/

[7] Vincent D Blondel, Jean-Loup Guillaume, Renaud Lambiotte, and Etienne Lefebvre. 2008. Fast Unfolding of Communities in Large Networks. *Journal of Statistical Mechanics: Theory and Experiment* 2008, 10 (2008), P10008.

[8] Károly Boda, Ádám Máté Földes, Gábor György Gulyás, and Sándor Imre. 2011. User tracking on the web via cross-browser fingerprinting. In *Nordic Conference on Secure IT Systems*. Springer, 31–46.

[9] A. Cahn, S. Alfeld, P. Barford, and S. Muthukrishnan. 2016. An Empirical Study of Web Cookies. In *Proceedings of the World Wide Web Conference (WWW '16)*. Montreal, Canada.

[10] SL Yinzhi Cao and E Wijmans. 2017. Browser Fingerprinting via OS and Hardware Level Features. In *Proceedings of the 2017 Network & Distributed System Security Symposium, NDSS*, Vol. 17.

[11] Electronic Privacy Information Center. 2017. Local Shared Objects – Flash Cookies. (October 2017). https://epic.org/privacy/cookies/flash.html

[12] CNBC. 2019. Apple Releases Privacy Commercial. (March 2019). https://www.cnbc.com/2019/03/14/apple-iphone-privacy-ad.html

[13] P. Eckersley. 2010. How Unique is Your Web Browser?. In *In Proceedings of the International Symposium on Privacy Enhancing Technologies*. Berlin, Germany.

[14] Rob van Eijk. 2019. Web privacy measurement in real-time bidding systems. A graph-based approach to RTB system classification. *A Graph-Based Approach to RTB System Classification (January 29, 2019). Web Privacy Measurement in Real-Time Bidding Systems. A Graph-Based Approach to RTB System Classification (diss. Leiden), Amsterdam: Ipskamp Printing* (2019).

[15] David Fifield and Serge Egelman. 2015. Fingerprinting web users through font metrics. In *International Conference on Financial Cryptography and Data Security*. Springer, 107–124.

[16] Keith Funkhouser, Matthew Malloy, Enis Ceyhun Alp, Philip Poon, and Paul Barford. 2018. Device Graphing by Example. In *Proceedings of the 24d ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 1913–1921.

[17] V. Goel. 2014. California Urges Websites to Disclose Online Tracking. *The New York Times* (May 2014).

[18] S. Kamkar. 2017. Evercookie. (October 2017). https://github.com/samyk/evercookie/commits/master

[19] B. Krishnamurthy, D. Malandrino, and C. Wills. 2007. Measuring Privacy Loss and the Impact of Privacy Protection in Web Browsing. In *Proceedings of the Symposium on Usable Privacy and Security*. Pittsburgh, PA.

[20] B. Krishnamurthy and C. Wills. 2006. Generating a Privacy Generating a Privacy Footprint on the Internet. In *Proceedings of the ACM Proceedings of the Internet Measurement Conference*. Rio de Janerio, Brazil.

[21] E. Lee. 2011. Sen. Rockefeller: Get Ready for a Real Do-Not-Track Bill for Online Advertising. *AdAge* (May 2011).

[22] Matthew Malloy, Paul Barford, Enis Ceyhun Alp, Jonathan Koller, and Adria Jewell. 2017. Internet Device Graphs. In *Proceedings of the 23rd ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. ACM, 1913–1921.

[23] Matthew Malloy, Mark McNamara, Aaron Cahn, and Paul Barford. 2016. Ad blockers: Global prevalence and impact. In *Proceedings of the 2016 Internet Measurement Conference*. ACM, 119–125.

[24] Matthew L Malloy, Scott Alfeld, and Paul Barford. 2015. Contamination estimation via convex relaxations. In *2015 IEEE International Symposium on Information Theory (ISIT)*. IEEE, 1189–1193.

[25] J. Mayer. 2011. Tracking the Trackers: Microsoft Advertising. (August 2011). http://cyberlaw.stanford.edu/blog/2011/08/tracking-trackers-microsoft-advertising

[26] J. Mayer and J. Mitchell. 2012. Third-Party Web Tracking: Policy and Technology. In *Proceedings of the IEEE Symposium on Security and Privacy*. San Francisco, CA.

[27] Jonathan R Mayer and John C Mitchell. 2012. Third-party web tracking: Policy and technology. In *2012 IEEE Symposium on Security and Privacy*. IEEE, 413–427.

[28] K. Mowery, D. Bogenreif, S. Yilek, and H. Shacham. 2011. Fingerprinting Information in JavaScript implementations. In *Proceedings of Web 2.0 Security and Privacy Workshop (W2SP)*. Oakland, CA.

[29] Keaton Mowery and Hovav Shacham. 2012. Pixel perfect: Fingerprinting canvas in HTML5. *Proceedings of W2SP* (2012), 1–12.

[30] Mozilla. 2017. BetterPrivacy. (2017). https://addons.mozilla.org/en-US/firefox/addon/betterprivacy/

[31] M. Murgia and D. Robinson. 2016. Google faces EU curbs on how it tracks users to drive adverts. *The Financial Times* (December 2016).

[32] N. Nikiforakis, A. Kapravelos, W. Joosen, C. Kruegel, F. Piessens, and G. Vigna. 2013. Cookieless Monster: Exploring the Ecosystem of Web-Based Device Fingerprinting. In *In Proceeding of the IEEE Symposium on Security and Privacy*. San Francisco, CA.

[33] Christopher Olston, Benjamin Reed, Utkarsh Srivastava, Ravi Kumar, and Andrew Tomkins. 2008. Pig latin: a not-so-foreign language for data processing. In *Proceedings of the 2008 ACM SIGMOD international conference on Management of data*. ACM, 1099–1110.

[34] Optanon. 2017. The Cookie Law Explained. (October 2017). https://www.cookielaw.org

[35] Franziska Roesner, Tadayoshi Kohno, and David Wetherall. 2012. Detecting and defending against third-party tracking on the web. In *Proceedings of the 9th USENIX conference on Networked Systems Design and Implementation*. USENIX Association, 12–12.

[36] Konstantinos Solomos, Panagiotis Ilia, Sotiris Ioannidis, and Nicolas Kourtellis. 2018. Cross-Device Tracking: Systematic Method to Detect and Measure CDT. *arXiv preprint arXiv:1812.11393* (2018).

[37] TAPAD. 2017. The TAPAD Device Graph. (October 2017). https://www.tapad.com/device-graph/

[38] Sebastian Zimmeck, Jie S Li, Hyungtae Kim, Steven M Bellovin, and Tony Jebara. 2017. A privacy analysis of cross-device tracking. In *26th {USENIX} Security Symposium ({USENIX} Security 17)*. 1391–1408.