The Representativeness of Automated Web Crawls as a Surrogate for Human Browsing

David Zeber, Sarah Bird, Camila Oliveira, Walter Rudametkin, Ilana Segall, Fredrik Wollsén, Martin Lopatka

To cite this version:

David Zeber, Sarah Bird, Camila Oliveira, Walter Rudametkin, Ilana Segall, et al.. The Representativeness of Automated Web Crawls as a Surrogate for Human Browsing. The Web Conference, Apr 2020, Taipei, Taiwan. 10.1145/3366423.3380104 . hal-02456195

HAL Id: hal-02456195
https://hal.inria.fr/hal-02456195

Submitted on 27 Jan 2020

The Representativeness of Automated Web Crawls as a Surrogate for Human Browsing

David Zeber
Mozilla
dzeber@mozilla.com

Sarah Bird
Mozilla
sbird@mozilla.com

Camila Oliveira
Mozilla
aliamcami@gmail.com

Walter Rudametkin
Univ. Lille / Inria
walter.rudametkin@univ-lille.fr

Ilana Segall
Mozilla
isegall@mozilla.com

Fredrik Wollsén
Mozilla
fred@augmentedmind.io

Martin Lopatka
Mozilla
mlopatka@mozilla.com

ABSTRACT

Large-scale Web crawls have emerged as the state of the art for studying characteristics of the Web. In particular, they are a core tool for online tracking research. Web crawling is an attractive approach to data collection, as crawls can be run at relatively low infrastructure cost and don't require handling sensitive user data such as browsing histories. However, the biases introduced by using crawls as a proxy for human browsing data have not been well studied. Crawls may fail to capture the diversity of user environments, and the snapshot view of the Web presented by one-time crawls does not reflect its constantly evolving nature, which hinders reproducibility of crawl-based studies. In this paper, we quantify the repeatability and representativeness of Web crawls in terms of common tracking and fingerprinting metrics, considering both variation across crawls and divergence from human browser usage. We quantify baseline variation of simultaneous crawls, then isolate the effects of time, cloud IP address vs. residential, and operating system. This provides a foundation to assess the agreement between crawls visiting a standard list of high-traffic websites and actual browsing behaviour measured from an opt-in sample of over 50,000 users of the Firefox Web browser. Our analysis reveals differences between the treatment of stateless crawling infrastructure and generally stateful human browsing, showing, for example, that crawlers tend to experience higher rates of third-party activity than human browser users on loading pages from the same domains.

CCS CONCEPTS

• **Information systems** → **Online advertising**; **Web mining**; **Traffic analysis**; *Data extraction and integration*; *Browsers.*

KEYWORDS

Web Crawling, Tracking, Online Privacy, Browser Fingerprinting, World Wide Web

ACM Reference Format:
David Zeber, Sarah Bird, Camila Oliveira, Walter Rudametkin, Ilana Segall, Fredrik Wollsén, and Martin Lopatka. 2020. The Representativeness of Automated Web Crawls as a Surrogate for Human Browsing. In *Proceedings of The Web Conference 2020 (WWW '20), April 20–24, 2020, Taipei, Taiwan.* ACM, New York, NY, USA, 12 pages. https://doi.org/10.1145/3366423.3380104

1 INTRODUCTION

The nature, structure, and influence of the Web have been subject to an overwhelming body of research. However, its rapid rate of evolution and sheer magnitude have outpaced even the most advanced tools used in its study [25, 44]. The expansive scale of the modern Web has necessitated increasingly sophisticated strategies for its traversal [6, 10]. Currently, computationally viable crawls are generally based on representative sampling of the portion of the Web most seen by human traffic [23, 47] as indicated by *top-site* lists such as Alexa [4] or Tranco [27].

Web crawls are used in a wide variety of research including fundamental research on the nature of the Web [32], training language models [21], social network analysis [33], and medical research [60]. In particular, they are a core tool in privacy research, having been used in numerous studies (see Section 2.2) describing and quantifying online tracking. However, little is known about the representativeness of such crawls compared to the experience of human users browsing the Web, nor about how much they depend on the environment from which they are run. In order to contextualize the information obtained from Web crawls, this study evaluates the efficacy, reliability, and generalisability of the use of Web crawlers [25] as a surrogate for user interaction with the Web in the context of online tracking. The main contributions of this paper are:

- A systematic exploration of the sources of variation between repeated Web crawls under controlled conditions
- Quantification of the variation attributable to the dynamic nature of the Web versus that resulting from client-specific environment and contextual factors

WWW '20, April 20–24, 2020, Taipei, Taiwan

- An assessment of the degree to which website ranking lists, often used to define crawl strategies, are representative of Web traffic measured over a set of real Web browsing clients
- A large-scale comparison between Web crawls and the experience of human browser users over a dataset of 30 million site visits across 50,000 users.

These results present a multifaceted investigation towards answering the question: when a crawler visits a specific page, is it served a comparable Web experience to that of a human user performing an analogous navigation?

2 BACKGROUND AND MOTIVATIONS

Seminal research into the graph-theoretical characteristics of the Web simplified its representation by assuming static relationships (edges) between static pages (nodes). Increasingly sophisticated Web crawlers [25] were deployed to traverse hyperlinkage structures, providing insights into its geometric characteristics [9]. Subsequent research has emphasized the importance of studying Internet topology [18, 31, 54]. Models accounting for the dynamic nature of the Web, and morphological consequences thereof [31], have encouraged emphasis on higher traffic websites in the context of an increasingly nonuniform and ever-growing Web. Applied research into specific characteristics of the Web [5], including the study of the tracking ecosystem that underlies online advertising [17, 50], have leveraged Web crawlers to study an immense and diverse ecosystem of Web pages.

Indeed, much of our understanding of how the Web works, and the privacy risks associated with it, result from research based on large-scale crawls. Yet, most crawls are performed only once, providing a snapshot view of the Web's inner workings. Moreover, crawls are most often performed using dedicated frameworks or specialized browsers implementations, differing significantly from the commodity browsing experience. Representativeness, repeatability, and variability are critical in such research, particularly in the study of tracking technologies, browser fingerprinting, or any number of security and privacy issues specific to the experience of human users. However, these issues have not been well studied in the context of large-scale crawl-based studies.

In this section we will define Web crawlers and provide an overview of the underlying technologies. We also look at some of the large-scale studies that rely on crawling, compare them to other sources of data, and conclude by discussing the issues that may introduce bias into crawls.

2.1 Crawl technology

A **crawler** is an automated client application that browses websites to collect data from them. The most well-known crawlers tend to be those used by search engines to index sites so as to make them easier to find. Other crawlers are used to gain a competitive advantage or to copy and republish data on third-party websites. Moreover, many Web measurement studies carried out by researchers rely on crawlers.

A **crawl** is the action of using a crawler to collect data from websites. Some crawls explore a single website, while in the case of large-scale studies, a single crawl may consist of millions of pages across several websites. The legal issues around crawling vary depending on the objectives of the crawl and are an active area of discussion [7, 48]. These are, however, outside the scope of this paper.

There are a multitude of crawler implementations that use different technologies with different tradeoffs. Simple crawlers are fast but do not run JavaScript, a central component of the modern Web environment. Other crawlers require more resources but are better at simulating a user's browser. The simplest forms of crawlers are scripts that use `curl` or `wget` to get a webpage's contents. More complete crawlers rely on frameworks like Scrapy [51], a Python-based framework for crawling and scraping data. Selenium [11] goes a step further by providing plugins to automate popular browsers, including Firefox, Chrome, Opera and Safari. Its popularity has led to the W3C Webdriver Standard [12]. In fact, Firefox and Chrome now provide headless modes that allow them to be run fully-automated from the command-line, without the need for a graphical interface. Libraries such as Puppeteer [20] provide convenient APIs to control browsers. Finally, OpenWPM [17] is a privacy measurement framework built on Firefox that uses Selenium for automation and provides hooks for data collection to facilitate large-scale studies.

There are important considerations when choosing a crawler technology. Simple crawlers that do not run JavaScript and are quite limited in modern dynamic environments. A stateful crawler, i.e., one that supports cookies, caches or other persistent data, may be desirable to better emulate users, but the results of the crawl may then depend on the accumulated state, e.g., the order of pages visisted. Finally, a distributed crawl may exploit the use of coordinated crawlers over multiple boxes with distinct IP addresses.

2.2 Web measurement studies

Crawls are a standard way of collecting data for Web measurements. Many crawl-based studies focus on security and privacy issues, such as online tracking [2, 17], detecting the use of browser fingerprinting [3, 26, 40], detecting the use of sensitive APIs [13], the tracking ecosystem of internet-enabled devices [34], dark patterns on shopping websites [28], content security violations [55], the effectiveness of tracker blockers [30], GDPR compliance [8], and many others [43, 57]. The OpenWPM framework has been used in over 47 such crawl-based studies [59]. While bias is often acknowledged in such studies, we are not aware of an exploration of how representative crawls are compared to users' experiences, how susceptible they can be to variability across IP addresses, regions, time, or how platforms or technologies may influence it.

Of course, crawls are not the only way to perform Web measurement studies. Other works have built datasets based on proxy interceptions [22, 24, 45], instrumented networks and VPNs [24, 45, 53]. Papadopoulos et al. use a dataset collected over 1 year from 850 participants and explicitly state they are not affected by distortions that exist in crawled data [45]. Another approach is to collect data through websites that provide information and feedback to users to entice them to participate [16, 26, 42], as well as apps and browser extensions [46, 61, 62]. Crowd-sourcing services can also be used to pay users to perform specific actions [29]. Yet, all of these approaches have their own biases and can lead to a lack of representativeness. Paying users introduces selection bias, as well as the possibility that user behavior is different when users know they are being monitored. Sites like AmIUnique [41] or Panopticlick

[16] have biased audiences that are already interested in the privacy issues they address, and are generally more technically proficient.

2.3 Bias and variability in Web crawls

There is plenty of anecdotal evidence describing issues surrounding Web crawls and scraping, as well as strategies to avoid detection [1, 56]. Specific robot detection strategies have been published which consider both the crawler's behavior (e.g., number of pages loaded) and originating IP address [15]. However, there is little information regarding the representativeness of crawls compared to human users' experiences, and on the variability that can be expected between simultaneous crawls, the variability of crawls over time, and the reproducibility of crawl-based research.

The technical articles written to improve crawlers or to avoid detection hint at some of the issues that can introduce variability or lack of representativeness e.g. [1]. These may include IP address reputation, including the use of cloud VMs, residential or business IP addresses, as well as the crawl technology, the browser and underlying platform, the region, the crawl's statefulness, and others. In this paper we provide a systematic exploration of the sources of variation between Web crawls and attempt to control the variation to as few factors as feasible. We also study the representativeness of crawls in regards to the experience of users on the Web.

3 METHODOLOGY

We approached our study of Web crawls by first analysing sets of repeated crawls under variable conditions, assessing the extent of crawl-to-crawl variation. Those findings then informed a systematic examination of the divergences observed between a large-scale Web crawl and browsing data obtained from actual Web users over the same time period. The source code and configuration for all our measurements and analysis is available publicly [39].

3.1 Measurement

All the Web measurements used in our analyses, from both crawl and user data, were collected using OpenWPM [17]. We captured webNavigations [58] (henceforth referred to as "navigations"), HTTP requests/responses/redirects, and JavaScript calls, the latter being limited to certain fingerprinting-related attributes and methods of the browser API. To collect data from live user browsing sessions, we integrated the OpenWPM instrumentation into a Firefox WebExtension and deployed it to an opt-in user population (further described in Section 3.4). Due to the verbose nature of the OpenWPM instrumentation and the scale of the data capture cohort, we group HTTP and JavaScript payloads by navigations and restrict data capture within each navigation group to 500 KiB[1], a 10-second cutoff or 1,000 events (whichever limit was hit first).

Additionally, to reduce known sources of bias between OpenWPM running as part of a WebExtension and crawlers run on different platforms, we contributed three significant enhancements to OpenWPM not available in previous studies:

- **Instrumentation of the webNavigation API**, which allows comparing crawl and user page loads consistently.
- **Support for Firefox Quantum (Firefox versions 57 and up)**, enabling data collection on recent Firefox versions, which account for a majority of users.[2]
- **Alternative orchestration:** The default OpenWPM orchestration tools (crawl setup, execution, data aggregation, etc.) currently support OSX and Linux. We built new orchestration tooling enabling crawls to be performed in a unified manner across Windows, OSX, and Linux.[3]

3.2 Definitions and metrics

We preprocess all URLs in our datasets by *stemming*, i.e., stripping all but their hostname and path components; this makes deduplication of URLs more meaningful. For example, stemming the URL https://x.y.com/path/a.html?a=1&b=2 gives x.y.com/path/a.html. Henceforth, **URL** refers to its stemmed version. We use the term **domain** as a shorthand for *eTLD+1*, the effective top-level domain plus one subdomain, sometimes known as the pay-level domain. For example, x.y.com and z.y.com share the same domain y.com.

A **site visit** refers to the notion of a user intentionally visiting a website, for example by clicking on a link, and encapsulates all related events such as requesting resources from other URLs and executing scripts. In the case of a crawl, this is the outcome of pointing the crawler to a specific URL. It is more tricky to determine top-level site visits in OpenWPM data collected from live user browsing; the approach we use is outlined in Section 3.5. Note that site visits are distinct from navigations, since separate navigation events are triggered for each frame. Typically, a site visit entails multiple navigations.

Finally, we say a resource is loaded from a **third-party domain** if the request domain is different from the domain of the site visit. Note that this can result in domains that are owned by the same entity being marked as third-party, such as a site visit to amazon.com loading images from ssl-images-amazon.com. Nevertheless, this is a common assumption in Web measurement literature [17, 50] and is contextualized by our additional tracking metrics.

With our focus on tracking and fingerprinting in mind, we use the following metrics for analysis. For each metric, we summarize events per site visit using aggregate counts, and compare sets of events between two site visits using Jaccard similarity, which offers a view into the evolution of third-party content on a per-site basis.

Third-party resources. We measure the prevalence of third-party content by counting the number of unique third-party domains or URLs across all HTTP requests generated by a site visit. We also consider the Jaccard similarity between sets of unique third-party domains or URLs loaded across two site visits.

Tracking resources. A third-party resource is considered to be "tracking" if its domain is on the Disconnect [14] list. We study the number of unique tracking domains for a site visit, as well as the Jaccard similarity between sets of unique tracking domains across two site visits.

Fingerprinting. We identify audio, Canvas, font, and WebRTC fingerprinting by Javascript scripts using published heuristics [13,

[1] Size limit applies to the encrypted combined payload that includes all of a specific navigation's HTTP and JavaScript payloads.

[2] The original OpenWPM was instrumented using Firefox APIs that were deprecated in the Quantum release of Firefox. All instrumentation was ported to work with the WebExtensions API.

[3] When alternative orchestration was used to run an experiment it has been noted. Otherwise default orchestration was used.

17]. We count the number of unique domains or URLs of scripts engaging in fingerprinting, as well as the Jaccard similarity between sets of unique fingerprinting script domains or URLs discovered globally across two crawl datasets.

Note that, for most of the analysis, we aggregate across all site visits under the same domain, in order make comparisons on the basis of site domains. This means that visits to subdomains of the same site will be grouped together.

3.3 Crawl-to-crawl comparisons

To quantify the natural variability inherent in the crawls themselves, we conducted a variety of experiments running multiple crawls with, as much as possible, all but one condition held constant. All of the crawls described in this section used the AlexaTop1k [4] as their seed list and were performed between July and October 2019. A page dwell time of 10 seconds was chosen to capture a majority of page load events for a majority of pages based on observations from comparable studies [19, 49], which we validated in our own preliminary exploration.

3.3.1 Baseline variation. We first investigated the variability observed under identical conditions: when the same crawler in the same operating system at the same IP address goes to the same website at approximately the same time. OpenWPM was run on a single Google cloud server, with 16 crawlers each simultaneously running a single instance of the Firefox 69 browser. We performed this for both the default and alternative orchestration versions of OpenWPM so that we could establish baselines for subsequent measurements depending on the technology configuration used. In both cases, one instance failed due to technical failure, resulting in successful data being collected from 15 simultaneous crawls.

3.3.2 Effect of time. 44 crawls using Firefox 68 were performed, with a variable cadence, over 54 days, using a single OSX machine located at a residential IP address in Brazil. Of these, 5 crawls failed to complete, leaving 39 crawls for analysis. The crawls were not run in headless mode, i.e., a normal browser window was opened for each crawled site.

3.3.3 Cloud vs. residential IP address. Crawls were run simultaneously on Linux at a residential IP address in Texas, USA and on a Google cloud server. Firefox 69 was used and was run on the in-memory display server Xvfb, a common tool for running browsers in servers for testing or crawling. Xvfb was also used for the residential IP address to limit variability in this experiment to IP address. Crawls were initiated simultaneously at both the cloud and residential locations. This procedure was repeated 4 times. The crawlers were configured to run with 3 parallel browser sessions running—that is, 3 browsers shared the crawl list and ran 1/3 each. These crawls used our alternative orchestration tooling.

3.3.4 Effect of operating system. Simultaneous crawls ran on Linux, OSX, and Windows at a single residential IP address in Texas, USA using Firefox 69. As in Section 3.3.3, each crawler ran 3 parallel browser sessions, and the procedure was repeated 4 times. The Linux crawl was run using Xvfb while OSX and Windows crawls were run with normal platform display (browser windows opening for each visited site). Using Linux-on-Xvfb in this experiment allows us to compare the cloud environment to user environments while controlling for variation that may be introduced by cloud vs residential IP address. These crawls were run using our alternative orchestration tooling.

3.4 Human-to-crawl comparisons

In order to quantify the representativeness of crawls in relation to the browsing experience of real human users, we collected contemporaneous datasets from both sources.

3.4.1 Human data collection. User traffic data was recorded using a WebExtension that bundled the OpenWPM instrumentation [36]. It was deployed to users of the Firefox browser who had opted in to participate in studies collecting data falling outside Mozilla's default data collection policies [37]. Approximately 52,000 users participated, all of whom were using Firefox versions 67 or 68 in the en-US localization at the time of data collection. We note here that due to the extremely sensitive nature of this dataset it is not feasible to publish it in its entirety with the rest of our data and analyses and therefore only selected analysis code and data summaries are available publicly [39].

The WebExtension collected user data over an initial period of 7 days, then paused data collection for 7 days, then resumed data collection for a second 7-day period, after which it uninstalled itself. These collection periods lasted for 7 calendar days, regardless of whether the user was active in the browser. A two-period collection scheme was employed to provide a compromise between data collection over a longer time period, which facilitates the study of longitudinal changes, and practical considerations regarding the volume of data collected and operational costs. The final dataset covers browsing which occurred between July 16 and August 13, 2019, and contains 30 million site visits covering 640,000 unique domains.

Note that our dataset was limited to an existing pool of opt-in users, all of whom had been part of this cohort for at least six months. Hence, while these users were not recruited into the cohort specifically for this study, they are likely not representative of general browser users across all dimensions. Nonetheless, our data provides a good approximation of general user behavior in terms of the particular metrics we elected to study, and aligns with previous work outlined in Section 2.2.

3.4.2 Companion crawl. As a basis for comparison, we launched a companion crawl on July 31, 2019. The crawl ran from Google cloud infrastructure and employed a page dwell time of 10 seconds. Obtaining crawl data that is comparable to real user browsing requires a judicious choice of the seed list of sites for the crawler to visit to ensure sufficient overlap with user browsing behaviour. Prior research has discussed the difficulty of appropriate website sampling in the context of Web crawls [6, 10]. We approached this problem by evaluating multiple top-site lists, and we settled on a hybrid list methodology.

We compared three well-known top-site lists—Alexa [4], Tranco [27], and Majestic Million [52]—and found them to differ substantially; in other words, they represent complementary sets of websites. The Jaccard similarity between Tranco and each of the other two over the top N ranks appears to converge to around 40% when N is increased beyond 100. The Majestic and Alexa lists exhibit more significant differences, with a similarity of around 20% over the same ranks. We then compared each list against aggregated site visits performed by actual users in a previously collected dataset of navigation events truncated to the top-level domain. This analysis revealed that retaining the top 10,000 domains from any of the individual lists attains an overlap of at least 40% with user site visits.

Informed by these findings, we decided to create a hybrid list combining entries across existing lists and including multiple Web properties belonging to each top-level domain. This accounts for the complementary nature of the standard lists, as well as the fact that human user browsing is better described by a hierarchy of pages across fewer top sites than by top-level home pages across a longer list of domains. We created a base list by interleaving the first 10,000 sites from each of the Alexa and Tranco top-site lists, dropping duplicate entries (the Majestic list added negligible additional information). The resulting list, which we call *Trexa10KU*, contains 14,556 domains. We then ran a depth-N pre-crawl of the Trexa10KU using a custom Scrapy [51] spider, recording at most 10 random visitable URLs linked from each main site. If fewer than 10 such links were encountered, a random link was followed and the same procedure was repeated until at most 10 visitable URLs were obtained. These URLs formed the final seed list, containing 108,207 entries. The main crawl, seeded with this list, resulted in 102,330 successfully visited URLs across 15,768 unique domains, of which 75% belong to the base Trexa10KU list.

3.5 Data preparation

Prior to the analysis, we performed some additional data processing steps to address inconsistencies between the crawl data and that collected from browser users. First, although the crawls were configured to use a dwell time of 10 seconds per page, the crawlers are implemented such that this is a lower bound, and may record well over 10 seconds worth of data on some pages. For example, certain pages appear to time out because the original `GET` request does not return correctly, but the page does actually load and generate data. In this case the crawler will have collected 60 seconds worth of data (the timeout threshold). To make comparisons consistent, we imposed a 10-second cutoff on all crawl datasets: for each site visit, we retained only those events which occurred within 10 seconds of the earliest recorded event.

The other major difference involves the notion of site visits. In a crawl, a site visit is well-defined: a new browser instance is spawned and navigates to a site. A "visit" then consists of all subsequent events. In the case of human user traffic, events may be interleaved across multiple concurrent site visits, and while the current implementation does group events by navigations, it does not associate navigation events with top-level intentional user site visits (recall that a site visit may generate multiple navigations, including frame loads which are considered separate navigations by OpenWPM).

To resolve this, we applied the following heuristic to group navigations into site visits. Within each browser session, window and tab, we identified each navigation event that *did not* take place in a frame (detected in the dataset as having a `frame_id` of 0) as a separate site visit. We then associated all subsequent frame navigation events prior to the next top-level navigation as belonging to the most recent site visit, with ordering given by the `before_navigate_event_ordinal` (which provides a sequential ordering of navigation events within each tab). Finally, we applied a visit-level 10-second cutoff by dropping any navigation events that occurred more than 10 seconds after the timestamp of the initial site visit navigation. While this heuristic relies on the assumption that each non-frame navigation represents a site visit, and ignores visit-related events that occur in other tabs such as pop-ups, we find it to be a reasonable compromise towards maximizing comparability given the constraints of our data collection. Quantifying the error this introduces would require significant further analysis which we leave for future work.

4 RESULTS

4.1 Crawl-to-crawl comparisons

In this section we report the results of our crawl-to-crawl comparisons and explore sources of variability. We break the metrics described in Section 3.2 into five sub-metrics: third-party domains, third-party URLs, tracking domains, fingerprinting script domains, and fingerprinting script URLs. For a top-level view into volume of content, we compute counts of each of these within each crawl and site visit, and compare the per-site distributions across crawls. To investigate changes in the content itself, we also compute the Jaccard similarity between pairs of crawls for each site, and compare the per-site distributions of Jaccard scores. We first establish a baseline for variability using simultaneous crawls, and then explore deviations relative to that baseline over three variables: time, IP address, and operating system.

4.1.1 Number of resources loaded. We reviewed the distributions of per-site resource counts and found very little variability between crawls for any of the measurements. For each crawl performed, we then computed the mean, median, and maximum values of each crawl's distribution over sites, and reported the average and standard deviation of these values across crawls in Table 1.

Due to the low variability, we do not explore these distributions of counts in further detail. We do, however, note the slightly higher standard deviations reported for the mean count of third-party URLs, third-party domains, and tracking domains in the case of OSX. This can be attributed to one of the four OSX crawls, which had a noticeably different distribution from the others. However, the remaining three were highly comparable, so we do not believe that OSX experiences systematically higher variability than other platforms.

4.1.2 Simultaneous crawls. We review the baseline distributions of Jaccard similarities across pairs of simultaneous crawls for each metric of interest, e.g., the similarity of the two sets of third-party domains loaded in visits to the same site between pairs of crawls.

In Figure 1 we plot the distributions over sites for third-party URLs. Each plot contains 105 lines, 1 for each pair of crawls. The tight, overlapping distributions show that the crawls exhibit repeatable, stable results while at the same time demonstrating the dynamism of the Web: the same third-party URLs are loaded only

Table 1: Distributional summaries of number of resources loaded, averaged over indicated crawls (standard deviation in parentheses). Distributions are per site, e.g., Windows crawls reported a median of 8.8 tracking domains per site on average with a standard deviation of 0.5. Corresponding summaries over all crawls are reported in bold.

		Mean	Median	Max
Tracking domains	Baseline	15.9 (0.38)	8.0 (0.00)	91.0 (1.43)
	Time Crawls	14.5 (0.32)	9.1 (0.27)	81.0 (4.31)
	Linux (cloud)	16.0 (0.38)	8.0 (0.00)	91.3 (1.89)
	Linux (local)	14.8 (0.76)	8.0 (0.00)	87.3 (8.77)
	OSX	13.4 (1.49)	7.8 (0.50)	88.8 (6.85)
	Windows	14.5 (0.76)	8.8 (0.50)	86.0 (5.48)
		15.0 (0.89)	**8.5 (0.59)**	**85.8 (6.10)**
Fingerprinting script domains	Baseline	1.07 (0.01)	1.00 (0.00)	2.37 (0.49)
	Time Crawls	1.05 (0.01)	1.00 (0.00)	2.00 (0.00)
	Linux (cloud)	1.06 (0.01)	1.00 (0.00)	2.25 (0.50)
	Linux (local)	1.07 (0.01)	1.00 (0.00)	2.75 (0.50)
	OSX	1.05 (0.02)	1.00 (0.00)	2.25 (0.50)
	Windows	1.07 (0.02)	1.00 (0.00)	2.50 (0.58)
		1.06 (0.01)	**1.00 (0.00)**	**2.21 (0.41)**
Fingerprinting script URLs	Baseline	1.40 (0.01)	1.00 (0.00)	4.53 (0.57)
	Time Crawls	1.36 (0.02)	1.00 (0.00)	3.03 (0.16)
	Linux (cloud)	1.38 (0.01)	1.00 (0.00)	4.25 (0.50)
	Linux (local)	1.41 (0.01)	1.00 (0.00)	5.00 (0.00)
	OSX	1.39 (0.02)	1.00 (0.00)	4.25 (0.50)
	Windows	1.40 (0.02)	1.00 (0.00)	4.00 (0.00)
		1.38 (0.02)	**1.00 (0.00)**	**3.80 (0.84)**
Third-party domains	Baseline	21.5 (0.38)	12.0 (0.09)	124.4 (4.26)
	Time Crawls	19.8 (0.39)	13.1 (0.33)	113.5 (5.70)
	Linux (cloud)	21.6 (0.46)	12.0 (0.00)	121.8 (2.22)
	Linux (local)	20.1 (0.99)	11.5 (0.58)	118.8 (14.20)
	OSX	18.6 (1.74)	11.5 (0.58)	125.8 (10.56)
	Windows	20.1 (0.93)	12.5 (0.58)	117.5 (8.50)
		20.5 (1.06)	**12.5 (0.69)**	**118.7 (7.88)**
Third-party URLs	Baseline	88.5 (1.92)	58.5 (0.89)	603.1 (77.49)
	Time Crawls	81.2 (1.77)	60.2 (1.08)	451.5 (29.27)
	Linux (cloud)	88.6 (1.56)	60.0 (0.82)	603.3 (22.29)
	Linux (local)	83.5 (3.63)	56.1 (2.17)	504.5 (56.39)
	OSX	76.7 (6.61)	54.6 (2.75)	436.8 (68.09)
	Windows	82.0 (3.77)	60.0 (2.83)	554.8 (31.56)
		84.0 (4.45)	**59.1 (1.91)**	**518.0 (90.16)**

28% of the time, and the typical overlap in third-party URLs is around 90%. The compact distribution of the lines indicates that this phenomenon of different URLs being loaded is repeatable and that these simultaneous crawls can serve as a baseline from which to compare our other variables.

4.1.3 Jaccard similarities. In Figure 2 we report the distributions of the mean Jaccard indexes across our different variables—time, cloud vs. residential IP address, and operating system—compared to the baseline distributions.

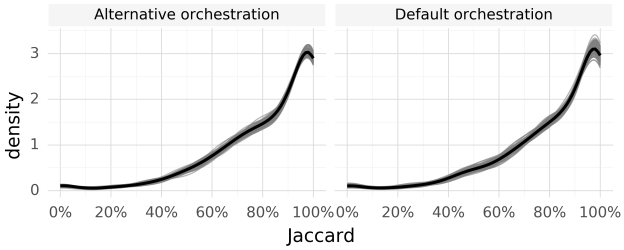

Figure 1: Distributions of pairwise Jaccard similarities per site for third-party URLs loaded during simultaneous crawls.

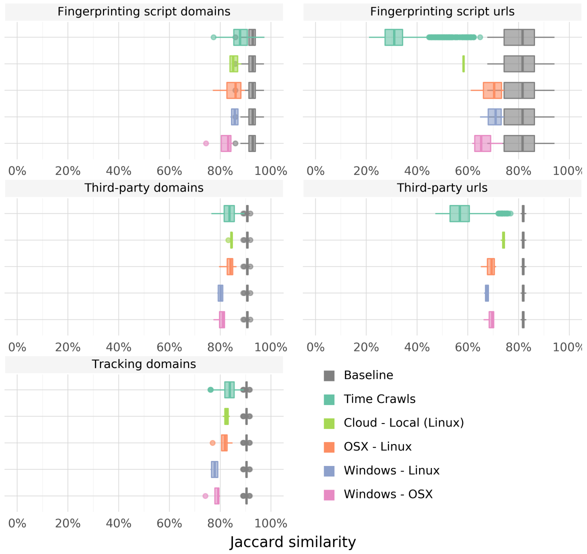

Figure 2: Distributions of mean Jaccard similarities between pairs of crawls for all metrics and crawl comparisons.

Overall, we find that the variability of URLs is higher than the variability of domains across our metrics. For the timed crawls, which have pairwise comparisons spanning from 1 to 54 days, there is a dramatic spread in the mean Jaccard similarity for both fingerprinting script URLs and third-party URLs. This difference is examined in detail in Section 4.1.4.

For our three operating system comparisons—OSX-Linux, Windows-Linux, and Windows-OSX—we observe that they are all different from baseline but not significantly different from one another. That is, being on a different platform has a measurable impact on the similarity of the resources you will be served, but no one platform comparison stands out as being dramatically different from any other. As just looking at means can obscure differences in the full range of results, we verified that this conclusion is the same when plotting the full distributions.

The Cloud-Local (Linux) comparison demonstrates the effect of IP address. We note that the size of this effect is generally smaller

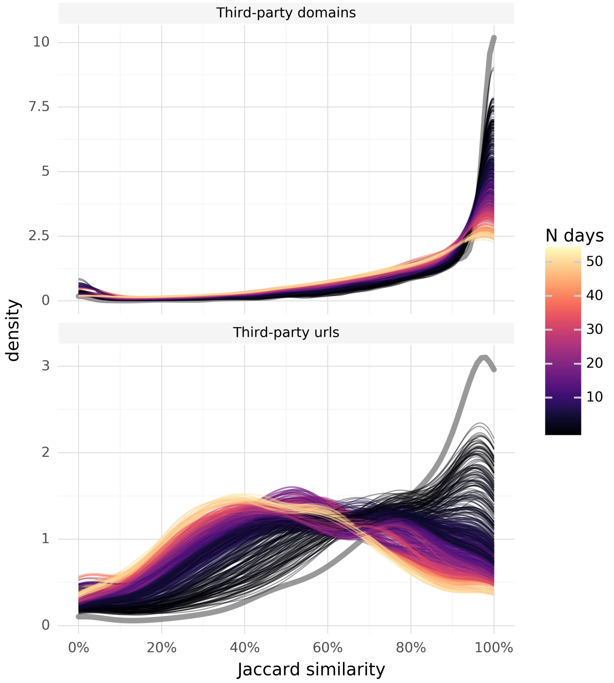

Figure 3: Distributions of Jaccard similarity for third-party URLs and domains over time. Baseline simultaneous crawls in gray.

than for operating system or time with the exception of fingerprinting script URLs.

For the domain metrics, the changes experienced over time are comparable to variations across operating system and IP address.

4.1.4 Variation over time. To further explore the effect of time on third-party resources, we plot the distributions of the Jaccard similarities for all pairs of crawls, shown in Figure 3. Each distribution is colored by the time between crawls, which range from 0.3 to 53 days. We see a clear trend: as the time between crawls decreases, the density approaches the shape of the baseline. The distributions of the Jaccard indexes quickly shift away from baseline and then stabilize, which can be seen in the tightening distance between distributions as the time difference increases. We explore this effect more closely in Figure 4 by looking at the distributions of the means of our five metrics split into discrete time windows. For each window, there is a noticeable drop in the Jaccard index compared to baseline. A notable example is fingerprinting script URLs over the 1-7 day time window. The median of this range is below 50%, indicating that crawls performed a week apart would expect less than half of their fingerprinting scripts to overlap, whereas the median of the baseline range is above 80%. Though this difference is particularly large, even less noteworthy drops in the Jaccard index will continue to compound over time to increase the difference between crawls. Work to explore the nature of this churn is out of the scope of this paper but would be valuable for understanding the nature of the tracking ecosystem.

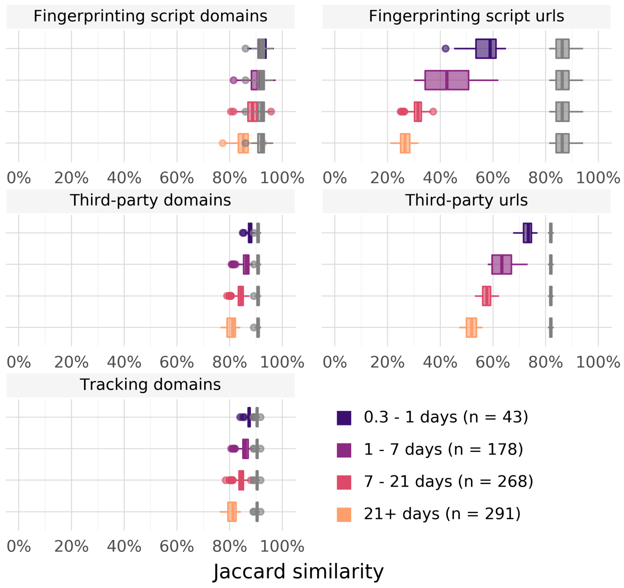

Figure 4: Distributions of mean Jaccard similarity between crawls, aggregated into time blocks

4.2 Consistency over time in human Web traffic

We now investigate whether we see similar effects over time in the human traffic dataset. In particular, we analyse the change in third-party domains accessed as a given user visits the same site on different days. For each user, we compute the Jaccard similarity between the third parties accessed on visiting each site domain on two different days. We then aggregate to obtain the mean Jaccard similarity score across all pairs of user visits that occurred N days apart for each site. The distribution of these scores across sites is displayed for each difference of N days in Figure 5, where we have split apart domains in the Trexa10KU list and other domains visited by the users. Since our dataset spans three weeks for each user, we are able to compute distributions for N ranging from 1 to 21. The top panel of this figure serves as a loose analogy from the user point of view to the time evolution for crawls presented in Figure 3. Although we find similar evidence in the user data of the leftward shift of the mode over time, the difference between the distributions for small lags is striking. For a day lag of 1, the mean Jaccard similarity over sites is only 40%, and few sites exhibit a similarity above 80%, a range that accounts for the vast majority of sites in the crawl data viewed over the same time lag. Interestingly, the distribution over sites not belonging to the Trexa10KU list show many more instances of both perfect agreement and no overlap, averaging at 20% over non-list sites compared to 5% in the list.

4.3 Comparison of crawls and human Web traffic

We begin by comparing the number of third-party domains accessed per site visit. We perform the comparison at the level of site visit domains, aggregating both human and crawl datasets to an

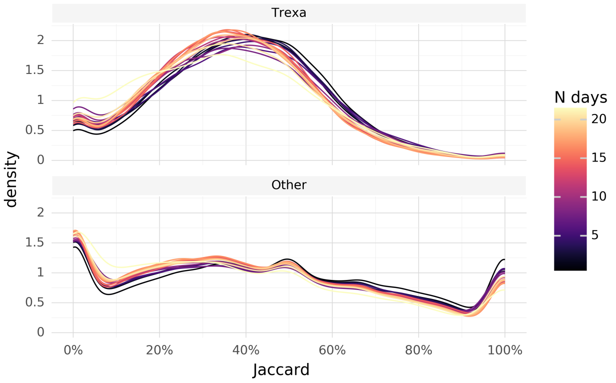

Figure 5: Distributions of Jaccard similarity for third-party domains within user and site visited across different date lags.

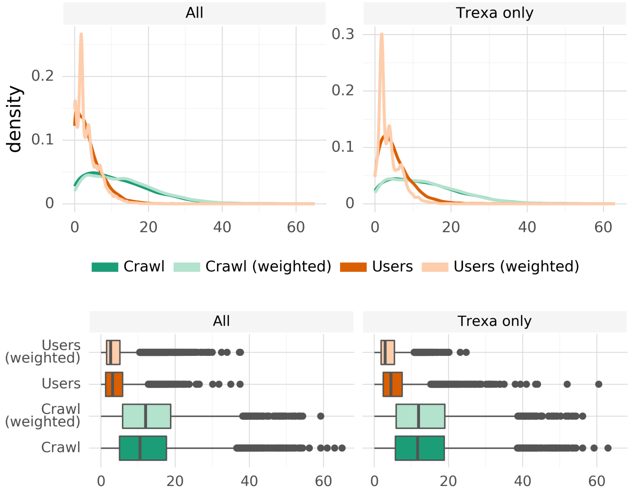

Figure 6: Distributions over visited domains of average number of third-party domains accessed

average number of third-party domains across all visits to a given domain. We then study the distribution of this value across unique domains. As well as including all domains reported in the datasets, we split out the subset of domains belonging to our base Trexa10KU list. We present two versions of each distribution, one where all domains are weighted equally, and a second where domains are re-sampled according to their relative popularity in terms of visit counts ("weighted"). This second view is more representative of a "typical" domain selected at random from the domains in the dataset. The distributions are shown in Figure 6.

We observe a compelling difference between the distributions, as the crawler tended to access significantly more third-party domains than users' browsers on the same sites. Among Trexa10KU domains, crawler site visits issued requests to a median of 11.6 third-party domains, whereas for visits by humans, the median was 4.5 third parties. Results are similar, albeit slightly lower, when taking into account all domains, suggesting that this difference is primarily driven by the top most popular sites. When considering the distributions weighted by popularity, the findings are similar in both cases, although the user distribution is shifted even lower, with the median dropping 35% to 2.9. Somewhat surprisingly, we found similar, although weaker, differences when applying the same analysis to the number of first-party requests.

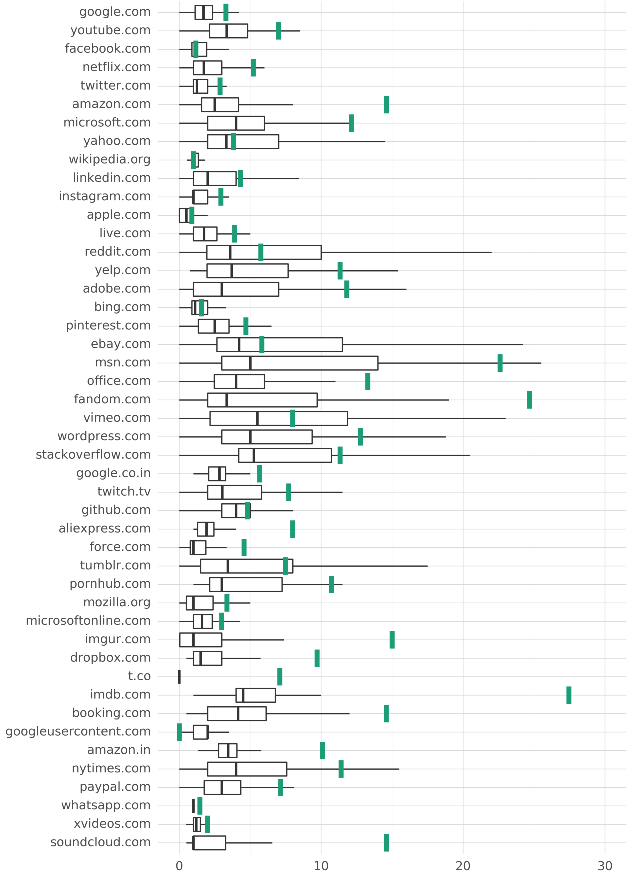

Figure 7: Distribution over users of average number of third-party domains accessed by visit domain (crawler value in green)

Figure 7 sheds further light on these results by splitting out individual top domains. For each domain, we compute the average number of third-party domains across visits by each individual user and represent the distribution across users in a boxplot. The value obtained by the crawler is annotated in green. The domains are selected as the subset of the top 100 Trexa10KU sites which were visited by at least 50 users, of which there are 46, ordered by their list rank. As well as displaying significant variation across user experiences, we find that for a majority of these top domains, the number of third parties reported by the crawler is well into the right tail of the distribution, in some cases even exceeding the maximum value reported by any individual user.

Next, we extend these findings by looking at the degree to which the actual third-party domains accessed differ between human

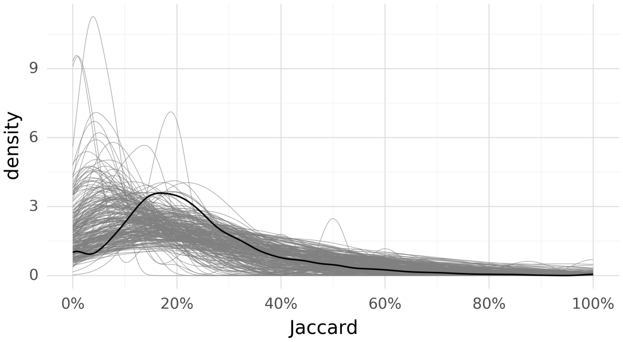

Figure 8: Distributions over visited domains of Jaccard similarity in third parties between individual users and crawl (per-visited-domain mean indicated in black)

browser users and the crawl. For each domain on the Trexa10KU list, we find the Jaccard similarity between the full set of third parties reached by each user on visiting sites at that domain and the corresponding set of third parties recorded by the crawler. In Figure 8 we display the distribution of average similarity values across sites, as well as the distributions over sites of the similarity values experienced by a sample of individual users. Overall, we find that the average similarity in third parties is low, with a median of 20%, and in most cases (87% of list domains) this is due to the crawler accessing more third parties. Approximately 4% of domains have no overlap whatsoever (Jaccard similarity of 0).

We now focus on third-party trackers and fingerprinters. Tracking domains are identified as those belonging to the Disconnect block list. We apply the same analysis as for third parties above to study the distributions of unique tracking domains to which requests were issued between human traffic and the crawl. Results are shown in Figure 9. The median number of tracking domains accessed by a user on visiting a Trexa list site is 1.9, whereas for the crawler it is 6.1. Furthermore, while users' browsers only connect to up to 8 trackers in 99% of visits to list sites, the crawler may reach 26!

Finally, we explore the extent of fingerprinting between the datasets. We identify fingerprinting scripts using a set of heuristics as outlined in Section 3.2. Figure 10 shows the prevalence of different types of fingerprinting across the site visit domains in our datasets, separating user traffic according to whether or not the visited site was on the Trexa10KU list. These results put occurrences of fingerprinting at around 12% of the Trexa sites, of which the majority is Canvas fingerprinting. In terms of fingerprinting scripts, there is a strong agreement between crawl and user data, with differences of at most 1% across all types. We also find fingerprinting to be much more prevalent among top sites than others, in accordance with previous literature. For each fingerprinting type, we also computed the Jaccard similarities in the domains and URLs from which the fingerprinting scripts are served between the datasets. Although we found very similar prevalence figures, this shows that, in fact, there is actually significant disparity between the sites serving fingerprinting scripts: domain overlap is in the 30-40% range for most fingerprinting types. Overlap between the actual script

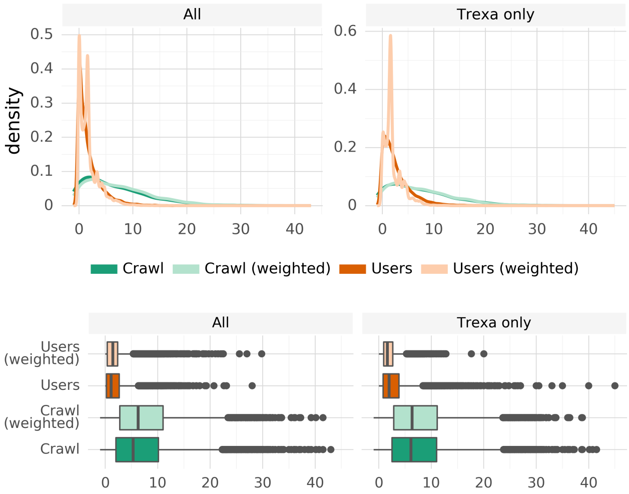

Figure 9: Distributions over visited domains of average number of known tracking domains accessed

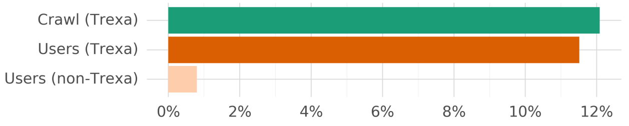

Figure 10: Prevalence of fingerprinting across visit domains

URLs is much lower. Based on these results, we believe that future crawl studies can be made more robust in terms of capturing the state of the Web by including multiple replicates, simultaneous or staggered over time, as appropriate. In particular, replicating crawls simultaneously is a relatively inexpensive way to bolster study results by providing an accompanying level of baseline variation.

5 DISCUSSION

The findings outlined in Section 4 provide a window into the dynamic nature of the Web. All crawl comparisons showed that similar *amounts* of content were being loaded, but the content itself was not identical, overlapping by only 90% in third-party domains and 81% in URLs. This was the case even between simultaneous crawls with identical configurations.

For each variable we examined (i.e., time, operating system, and cloud vs residential IP address), there was a reduction in similarity of the content being loaded relative to the baseline. Domains were more similar than URLs, which is unsurprising as it is much simpler and cheaper to dynamically change or update URLs compared to domains. Of the decreases in similarity, time had a profound effect for the similarity of URLs: in the most stark case, after three weeks only 25% of fingerprinting script URLs were the same. Future research should investigate this effect to better understand the limitations of crawls for measuring the fingerprinting landscape.

There was a noteworthy effect of cloud vs. residential IP address in fingerprinting script URL similarity. Although establishing a cause for this is out of scope for this paper, we note that fingerprinting can be used for anti-fraud and bot detection purposes [38]. As such, crawls coming from cloud IP addresses may be flagged as potential bots and served different fingerprinting scripts as a mitigation.

We also compared human traffic data with a companion crawl. Our results provide new insights that should be taken into account when extrapolating results from crawl studies to user experiences. For example, although there were only minor differences in the overall prevalence of fingerprinting across top sites, the specific domains hosting fingerprinting scripts exhibited low similarity between human and crawl data.

In Section 4.3 we find that crawlers tend to experience more third-party activity than when the same sites are visited by humans, with the median number of tracking domains increasing threefold. The high difference in the amount of traffic limited our ability to bring additional insight by looking at the Jaccard similarity between crawls and humans. Due to the observational nature of data originating from client sessions, causal mechanisms responsible for such variations could not be studied directly. While the differential handling of automated crawler traffic has been raised as a potential issue, our crawl-to-crawl analysis (e.g. Section 4.1.3) provides some evidence that this may play a relatively minor role. We note that prior research has shown cookie syncing to be aggressive in the number of HTTP requests it generates and that cookie syncing is not necessary for users who have already had their cookies synced, whereas a stateless crawler browser instance with a fresh profile would be a clear target for cookie syncing. Englehardt et al. [17] demonstrated that third-party cookie blocking dropped the average number of third parties from 17.7 to 12.6, a finding comparable with our results. Additionally, during the time of our data collection period, cross-site-tracker cookie blocking was being deployed to Firefox users [35]. Further analysis revealed that crawlers also triggered a higher number of requests to first-party domains compared to users. One interpretation is first-party participation in the advertising and tracking ecosystem, which would lead to similar interactions with cookies as with third parties. We believe these findings merit further exploration in future work.

In Figure 5 we see that when the same user visits the same site, even on the same day, the average similarity in third-party domains is low. This is dissimilar to our crawl-to-crawl comparison where we see an evolution of reducing similarity over time. It seems that this result should temper the previous finding that users are exposed to fewer third parties. Users may be exposed to fewer third parties on any one visit, but if they are different each time, it might not be long before they've accumulated many different third parties. One possible explanation is that users' interactions on a domain are more heterogeneous compared to a crawler. Another is the complex nature of the ad tech ecosystem: when a page is loaded, numerous decisions are made about the type of ad shown to a user, including the means of display, such as whether or not to participate in a real time bidding auction [63]. Therefore, crawlers, which have not accumulated advertising profiles, may be treated systematically differently compared to humans. Exploring the dissimilarity of third parties was out of scope for this paper. However, future work should investigate this carefully to enable researchers to more accurately represent the exposure of users to third parties in a crawl.

In general, there are a multitude of reasons why data collected by a crawler may not be representative of the actual human browsing experience. Crawls may fail to capture the diversity of user environments, including operating systems, geolocation, and profile history. Users may also be seeing different portions of websites, such as content specific to their interests (personalized or self-selected) or content only visible in authenticated sessions or behind paywalls, or they may be getting different localized content depending on their geographic location. These sources of bias are inherent in crawl research and difficult to design around. However, we believe our results present a novel, pragmatic view into the aggregate effect of these differences, and that future work could help to quantify this phenomenon and provide potential solutions.

6 CONCLUSION

Web crawling is an essential tool in the study of the Web and offers many advantages, not least circumventing the privacy issues inherent in collecting human user data. However, we believe the results obtained by crawling can be made even more compelling when contextualized in terms of the fundamental variation inherent in the Web and across the various user environments. Through the numerous results presented above, we dive into an area with significant implications for crawl-based Web research that has not previously received much attention, namely the repeatability and representativeness of crawl studies. We quantify the variability that crawl data collection is subject to over time and across platforms, as well as the baseline variation between identical crawls. We then study the biases involved in associating crawl results with actual user browsing.

The work we present here also raises a substantial number of questions for further research. While we observe interesting patterns of variation across a number of variables, the factors driving these results are not yet well understood. Their exploration constitutes an important direction for future work. Additionally, the OpenWPM measurement framework produces a very rich dataset, of which we have only scratched the surface in this work. There are many other features of interest to which our methodology could be extended, and we look forward to future research on this topic.

ACKNOWLEDGMENTS

The authors thank the security engineering team at Mozilla, especially Steven Englehardt for support in implementing necessary OpenWPM enhancements. This work is partially funded by the ANR FP-Locker project under grant number ANR-19-CE39-00201.

REFERENCES

[1] Scrape Hero: a data company. 2019. How to prevent getting blacklisted while scraping. https://www.scrapehero.com/how-to-prevent-getting-blacklisted-while-scraping. Accessed: 29-July-2019.

[2] Gunes Acar, Christian Eubank, Steven Englehardt, Marc Juarez, Arvind Narayanan, and Claudia Diaz. 2014. The Web Never Forgets: Persistent Tracking Mechanisms in the Wild. In *Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security (CCS '14)*. ACM, New York, NY, USA, 674–689. https://doi.org/10.1145/2660267.2660347

[3] Gunes Acar, Marc Juarez, Nick Nikiforakis, Claudia Diaz, Seda Gürses, Frank Piessens, and Bart Preneel. 2013. FPDetective: dusting the web for fingerprinters. In *Proceedings of the 2013 ACM SIGSAC conference on Computer & communications security*. ACM, ACM, Berlin, Germany, 1129–1140.

[4] Alexa: an Amazon.com company. 2019. Alexa: the top sites on the web. https://www.alexa.com/topsites. Accessed: 29-July-2019.

[5] Calvin Ardi and John Heidemann. 2019. Precise Detection of Content Reuse in the Web. *SIGCOMM Comput. Commun. Rev.* 49, 2 (May 2019), 9–24.

[6] Ricardo Baeza-Yates, Carlos Castillo, Mauricio Marin, and Andrea Rodriguez. 2005. Crawling a Country: Better Strategies Than Breadth-first for Web Page Ordering. In *Special Interest Tracks and Posters of the 14th International Conference on World Wide Web (WWW ’05)*. ACM, New York, NY, USA, 864–872. https://doi.org/10.1145/1062745.1062768

[7] Benoit Bernard. 2018. Web Scraping and Crawling Are Perfectly Legal, Right? https://benbernardblog.com/web-scraping-and-crawling-are-perfectly-legal-right/

[8] Coline Boniface, Imane Fouad, Nataliia Bielova, Cédric Lauradoux, and Cristiana Santos. 2019. Security Analysis of Subject Access Request Procedures. In *Privacy Technologies and Policy*, Maurizio Naldi, Giuseppe F. Italiano, Kai Rannenberg, Manel Medina, and Athena Bourka (Eds.). Springer International Publishing, Cham, 182–209.

[9] Andrei Broder, Ravi Kumar, Farzin Maghoul, Prabhakar Raghavan, Sridhar Rajagopalan, Raymie Stata, Andrew Tomkins, and Janet Wiener. 2000. Graph structure in the Web. *Computer Networks* 33, 1 (2000), 309 – 320.

[10] Carlos Castillo, Mauricio Marin, Andrea Rodriguez, and Ricardo Baeza-Yates. 2004. Scheduling algorithms for Web crawling. In *WebMedia and LA-Web, 2004. Proceedings*. IEEE, Ribeirao Preto, Brazil, 10–17.

[11] Software Freedom Conservancy. 2019. SeleniumHQ Browser Automation. https://selenium.dev/.

[12] World Wide Web Consortium. 2019. W3C Webdriver Standard. https://w3c.github.io/webdriver/

[13] Anupam Das, Gunes Acar, Nikita Borisov, and Amogh Pradeep. 2018. The Web’s Sixth Sense: A Study of Scripts Accessing Smartphone Sensors. In *Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security - CCS ’18*. ACM Press, Toronto, Canada, 1515–1532. https://doi.org/10.1145/3243734.3243860

[14] Disconnect. 2019. Disconnect Tracking Protection List. https://github.com/disconnectme/disconnect-tracking-protection

[15] Derek Doran and Swapna S. Gokhale. 2011. Web robot detection techniques: overview and limitations. *Data Mining and Knowledge Discovery* 22, 1 (01 Jan 2011), 183–210. https://doi.org/10.1007/s10618-010-0180-z

[16] Peter Eckersley. 2010. How Unique Is Your Web Browser?. In *Privacy Enhancing Technologies* (2010-07-21) *(Lecture Notes in Computer Science)*. Springer, Berlin, Heidelberg, Berlin, Germany, 1–18. https://doi.org/10.1007/978-3-642-14527-8_1

[17] Steven Englehardt and Arvind Narayanan. 2016. Online Tracking: A 1-million-site Measurement and Analysis. In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security (CCS ’16)*. ACM, New York, NY, USA, 1388–1401.

[18] Michalis Faloutsos, Petros Faloutsos, and Christos Faloutsos. 1999. On Power-law Relationships of the Internet Topology. *SIGCOMM Comput. Commun. Rev.* 29, 4 (Aug. 1999), 251–262.

[19] Mohammad Ghasemisharif, Peter Snyder, Andrius Aucinas, and Benjamin Livshits. 2019. SpeedReader: Reader Mode Made Fast and Private. In *The World Wide Web Conference (WWW ’19)*. ACM, New York, NY, USA, 526–537. https://doi.org/10.1145/3308558.3313596

[20] Google. 2019. Puppeteer. https://pptr.dev/

[21] Edouard Grave, Piotr Bojanowski, Prakhar Gupta, Armand Joulin, and Tomas Mikolov. 2018. Learning Word Vectors for 157 Languages. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Nicoletta Calzolari (Conference chair), Khalid Choukri, Christopher Cieri, Thierry Declerck, Sara Goggi, Koiti Hasida, Hitoshi Isahara, Bente Maegaard, Joseph Mariani, HÃĺlÃĺne Mazo, Asuncion Moreno, Jan Odijk, Stelios Piperidis, and Takenobu Tokunaga (Eds.). European Language Resources Association (ELRA), Miyazaki, Japan.

[22] David Gugelmann, Markus Happe, Bernhard Ager, and Vincent Lenders. 2015. An Automated Approach for Complementing Ad Blockers’ Blacklists. *Proceedings on Privacy Enhancing Technologies* 2015, 2 (2015), 282–298. https://doi.org/10.1515/popets-2015-0018

[23] Felix Hernández-Campos, Kevin Jeffay, and F.D. Smith. 2003. Tracking the evolution of Web traffic: 1995-2003. In *in: Proceedings of the 11th IEEE/ACM International Symposium on Modeling, Analysis and Simulation of Computer Telecommunication Systems (MASCOTS*. IEEE/ACM International, Orland, Florida, USA, 16–25. https://doi.org/10.1109/MASCOT.2003.1240638

[24] Vasiliki Kalavri, Jeremy Blackburn, Matteo Varvello, and Konstantina Papagiannaki. 2016. Like a Pack of Wolves: Community Structure of Web Trackers. In *Passive and Active Measurement* (2016-03-31). Springer, Cham, Heraklion, Crete, Greece, 42–54.

[25] Manish Kumar, Rajesh Bhatia, and Dhavleesh Rattan. 2017. A survey of Web crawlers for information retrieval. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery* 7, 6 (2017), e1218.

[26] Pierre Laperdrix, Walter Rudametkin, and Benoit Baudry. 2016. Beauty and the Beast: Diverting Modern Web Browsers to Build Unique Browser Fingerprints. In *2016 IEEE Symposium on Security and Privacy (SP)* (2016-05). IEEE, San Jose, CA, 878–894. https://doi.org/10.1109/SP.2016.57

[27] V. Le Pochat, T. Van Goethem, S. Tajalizadehkhoob, M. Korczyński, and W. Joosen. 2018. Tranco: A Research-Oriented Top Sites Ranking Hardened Against Manipulation. arXiv:cs.CR/1806.01156

[28] Arunesh Mathur, Gunes Acar, Michael Friedman, Elena Lucherini, Jonathan R. Mayer, Marshini Chetty, and Arvind Narayanan. 2019. Dark Patterns at Scale: Findings from a Crawl of 11K Shopping Websites. *CoRR* abs/1907.07032 (2019), 1–32. arXiv:1907.07032 http://arxiv.org/abs/1907.07032

[29] Wei Meng, Ren Ding, Simon P. Chung, Steven Han, and Wenke Lee. 2016. The Price of Free: Privacy Leakage in Personalized Mobile In-App Ads. In *Proceedings 2016 Network and Distributed System Security Symposium*. Internet Society, San Diego, California, USA. https://doi.org/10.14722/ndss.2016.23353

[30] Georg Merzdovnik, Markus Huber, Damjan Buhov, Nick Nikiforakis, Sebastian Neuner, Martin Schmiedecker, and Edgar Weippl. 2017. Block me if you can: A large-scale study of tracker-blocking tools. In *2017 IEEE European Symposium on Security and Privacy (EuroS&P)*. IEEE, IEEE, Paris, France, 319–333.

[31] Panagiotis Metaxas. 2012. Why Is the Shape of the Web a Bowtie?. In *Proceedings of the 2012 World Wide Web conference, WebScience Track (WWW ’12)*. IW3C2, Geneva, Switzerland, 6.

[32] Robert Meusel, Sebastiano Vigna, Oliver Lehmberg, and Christian Bizer. 2015. The Graph Structure in the Web - Analyzed on Different Aggregation Levels. *The Journal of Web Science* 1, 1 (2015), 33–47. https://doi.org/10.1561/106.00000003

[33] Alan Mislove, Massimiliano Marcon, Krishna P. Gummadi, Peter Druschel, and Bobby Bhattacharjee. 2007. Measurement and Analysis of Online Social Networks. In *Proceedings of the 7th ACM SIGCOMM Conference on Internet Measurement (IMC ’07)*. Association for Computing Machinery, New York, NY, USA, 29–42. https://doi.org/10.1145/1298306.1298311

[34] Hooman Mohajeri Moghaddam, Gunes Acar, Ben Burgess, Arunesh Mathur, Danny Yuxing Huang, Nick Feamster, Edward W. Felten, Prateek Mittal, and Arvind Narayanan. 2019. Watching You Watch: The Tracking Ecosystem of Over-the-Top TV Streaming Devices. In *Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security (CCS ’19)*. Association for Computing Machinery, New York, NY, USA, 131–147. https://doi.org/10.1145/3319535.3354198

[35] Mozilla. 2019. Firefox Now Available with Enhanced Tracking Protection by Default Plus Updates to Facebook Container, Firefox Monitor and Lockwise. https://blog.mozilla.org/blog/2019/06/04/firefox-now-available-with-enhanced-tracking-protection-by-default/. Accessed: 14-Oct-2019.

[36] Mozilla. 2019. JESTr Pioneer Shield Study. https://github.com/mozilla/jestr-pioneer-shield-study.

[37] Mozilla. 2019. Mozilla Privacy Policy. https://www.mozilla.org/en-US/privacy/

[38] Mozilla. 2019. Security/Anti tracking policy. https://wiki.mozilla.org/Security/Anti_tracking_policy#2._Tracking_via_unintended_identification_techniques. Accessed: 29-July-2019.

[39] Mozilla. 2019. Study Companion Repository. https://github.com/mozilla/research-repo-webconf-crawl-representativeness.

[40] 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 *2013 IEEE Symposium on Security and Privacy*. IEEE Computer Society, San Francisco, California, USA, 541–555. https://doi.org/10.1109/SP.2013.43

[41] Inria & University of Lille. 2019. AmIUnique. https://amiunique.org/.

[42] Lukasz Olejnik, Claude Castelluccia, and Artur Janc. 2012. Why Johnny Can’t Browse in Peace: On the Uniqueness of Web Browsing History Patterns. In *5th Workshop on Hot Topics in Privacy Enhancing Technologies (HotPETs 2012)*. De Gruyter, Vigo, Spain, 1–17. https://hal.inria.fr/hal-00747841

[43] Rebekah Overdorf, Mark Juarez, Gunes Acar, Rachel Greenstadt, and Claudia Diaz. 2017. How Unique is Your .Onion?: An Analysis of the Fingerprintability of Tor Onion Services. In *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security (CCS ’17)*. ACM, New York, NY, USA, 2021–2036. https://doi.org/10.1145/3133956.3134005

[44] T. K. Panum, R. R. Hansen, and J. M. Pedersen. 2019. Kraaler: A User-Perspective Web Crawler. In *2019 Network Traffic Measurement and Analysis Conference (TMA)*. ACM SIIGCOMM, Paris, France, 153–160.

[45] Panagiotis Papadopoulos, Nicolas Kourtellis, and Evangelos P. Markatos. 2018. Cookie Synchronization: Everything You Always Wanted to Know But Were Afraid to Ask. *arXiv e-prints* v2 (2018), 1–11. arXiv:cs/1805.10505 http://arxiv.org/abs/1805.10505

[46] Javier Parra-Arnau, Jagdish Prasad Achara, and Claude Castelluccia. 2017. MyAdChoices: Bringing Transparency and Control to Online Advertising. *ACM Transactions on the Web (TWEB)* 11, 1 (2017), 7:1–7:47. https://doi.org/10.1145/2996466

[47] Victor Le Pochat, Tom Van Goethem, and Wouter Joosen. 2019. Evaluating the Long-term Effects of Parameters on the Characteristics of the Tranco Top Sites Ranking. In *12th USENIX Workshop on Cyber Security Experimentation and Test (CSET 19)*. USENIX Association, Santa Clara, CA. https://www.usenix.org/conference/cset19/presentation/lepochat

[48] Quora. 2018. Is scraping and crawling to collect data illegal? https://www.quora.com/Is-scraping-and-crawling-to-collect-data-illegal

[49] A. Saverimoutou, B. Mathieu, and S. Vaton. 2019. Web View: Measuring Monitoring Representative Information on Websites. In *2019 22nd Conference on Innovation in Clouds, Internet and Networks and Workshops (ICIN)*, Vol. 4. ACM SIGCOMM Computer Communication Review, Paris, France, 133–138. https://doi.org/10.1109/ICIN.2019.8685876

[50] Sebastian Schelter and Jerome Kunegis. 2018. On the Ubiquity of Web Tracking: Insights from a Billion-Page Web Crawl. *The Journal of Web Science* 4, 4 (2018), 53–66. https://doi.org/10.1561/106.00000014

[51] Scrapinghub. 2019. Scrapy. https://scrapy.org

[52] Majestic SEO. 2019. The Majestic Million: The million domains we find with the most referring subnets. https://majestic.com/reports/majestic-million. Accessed: 29-July-2019.

[53] Anastasia Shuba, Athina Markopoulou, and Zubair Shafiq. 2018. NoMoAds: Effective and Efficient Cross-App Mobile Ad-Blocking. *Proceedings on Privacy Enhancing Technologies* 2018, 4 (2018), 125–140. https://doi.org/10.1515/popets-2018-0035

[54] Georgos Siganos, Sudhir Tauro, and Michalis Faloutsos. 2005. Jellyfish: A conceptual model for the AS internet topology. *Journal of Communications and Networks - JCN* 8 (01 2005), 1667–1671. https://doi.org/10.1109/JCN.2006.6182774

[55] Dolière Francis Some, Nataliia Bielova, and Tamara Rezk. 2017. On the Content Security Policy Violations Due to the Same-Origin Policy. In *Proceedings of the 26th International Conference on World Wide Web (WWW '17)*. International World Wide Web Conferences Steering Committee, Republic and Canton of Geneva, Switzerland, 877–886. https://doi.org/10.1145/3038912.3052634

[56] Nikolai Tschacher. 2019. Scraping 1 million keywords on the Google Search Engine. https://incolumitas.com/2019/09/17/scraping-one-million-google-serps. Accessed: 13-October-2019.

[57] Tom Van Goethem, Victor Le Pochat, and Wouter Joosen. 2019. Mobile Friendly or Attacker Friendly?: A Large-scale Security Evaluation of Mobile-first Websites. In *Proceedings of the 2019 ACM Asia Conference on Computer and Communications Security (Asia CCS '19)*. ACM, New York, NY, USA, 206–213. https://doi.org/10.1145/3321705.3329855

[58] MDN web docs contributors. 2019. webNavigation. https://developer.mozilla.org/en-US/docs/Mozilla/Add-ons/WebExtensions/API/webNavigation. Accessed: 29-July-2019.

[59] Princeton University WebTAP research group. 2019. Studies using OpenWPM. https://webtap.princeton.edu/software/

[60] Hao Wu, Hui Fang, and Steven J. Stanhope. 2012. An Early Warning System for Unrecognized Drug Side Effects Discovery. In *Proceedings of the 21st International Conference on World Wide Web (WWW '12 Companion)*. Association for Computing Machinery, New York, NY, USA, 437–440. https://doi.org/10.1145/2187980.2188068

[61] Zhonghao Yu, Sam Macbeth, Konark Modi, and Josep M. Pujol. 2016. Tracking the Trackers. In *Proceedings of the 25th International Conference on World Wide Web (WWW '16)*. International World Wide Web Conferences Steering Committee, Republic and Canton of Geneva, CHE, 121–132. https://doi.org/10.1145/2872427.2883028

[62] 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)*. USENIX Association, Vancouver, BC, 1391–1408. https://www.usenix.org/conference/usenixsecurity17/technical-sessions/presentation/zimmeck

[63] Aram Zucker-Scharff. 2019. Understanding the Unplanned Internet - How Ad Tech is Broken By Design 101. https://youtu.be/QIyxmSfKGbw?t=7907