Measurement-based Experiments on the Mobile Web: A Systematic Mapping Study

Omar de Munk
Vrije Universiteit Amsterdam, The Netherlands
o.de.munk@student.vu.nl

Ivano Malavolta
Vrije Universiteit Amsterdam, The Netherlands
i.malavolta@vu.nl

ABSTRACT

The mobile Web is growing as more and more people use a smart device to access online services. This rapid growth of mobile Web usage is accompanied by the evolution of the mobile Web browser as a fully fledged software platform. Due to these two trends, the expectations of users in terms of quality of experience (QoE) when browsing the Web on their mobile device has increased drastically. As a result, the number of studies using measurement-based experiments to investigate the factors influencing QoE has grown.

However, conducting measurement-based experiments on the mobile Web is not a trivial task as it requires a significant experience and knowledge about both technical and methodological aspects. Unfortunately, there is no systematic study on the state of the art of conducting measurement-based experiments on the mobile Web that could guide researchers and practitioners when planning and performing such experiments.

The goal of this work is to build a map of existing studies that conduct measurement-based experiments on the mobile Web. In total 640 potentially relevant studies are identified. After a rigorous selection procedure the set of primary studies consists of 28 papers from which we extracted data and gathered insights. Specifically, we investigate on (i) which metrics are collected, how they are measured, and how they are analysed, (ii) the platforms on which the experiments are run, (iii) what subjects are used, and (iv) the used tools and environments under which the experiments are run.

This study benefits researchers and practitioners by presenting common techniques, empirical practices, and tools to properly conduct measurement-based experiments on the mobile Web.

CCS CONCEPTS

• **Software and its engineering → Software performance**; **Empirical software validation**; • **Information systems → Web applications**;

ACM Reference Format:
Omar de Munk and Ivano Malavolta. 2021. Measurement-based Experiments on the Mobile Web: A Systematic Mapping Study . In *Evaluation and Assessment in Software Engineering (EASE 2021), June 21–23, 2021, Trondheim, Norway.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3463274.3463318

EASE 2021, June 21–23, 2021, Trondheim, Norway
© 2021 Association for Computing Machinery.
ACM ISBN 978-1-4503-9053-8/21/06…$15.00
https://doi.org/10.1145/3463274.3463318

1 INTRODUCTION

The mobile Web is ever growing as more and more people use a smart device to access the world wide Web. During the month November 2020, more than 55% of all worldwide Web traffic came from mobile devices. This is in contrast with the same period in 2015, when this percentage was only 42% [29]. For a large group of people their mobile device's Web browser functions as one of the major gateways to the internet.

The rapid growth of the mobile Web is accompanied by the evolution of the mobile browser as a fully-featured, almost operating system like, software platform thanks to the continuous development of the HTML5 specification. As a result mobile Web browsers now have APIs for geolocation, motion sensors and vibrations [7]. Due to these two trends, the expectations of users in terms of Quality of Experience (QoE) has increased drastically when browsing the Web on their mobile device. Various sources have shown the impact of QoE in terms of revenue and user retention. For example, Amazon calculated that a page load slowdown of just one second could cost them $1.6 billion in sales each year [9].

Since the quality of service can be affected by a wide variety of factors improving it is considered to be a non-trivial task. As a result the number of studies investigating these factors by doing measurement-based experiments is growing. This is important because the number of factors contributing to QoE is different than with desktop based systems when Web browsing on mobile devices. For example, energy consumption plays a much more critical role on mobile devices as they are equipped with a battery and often only have intermittent access to electricity grid. In addition, there is the importance of bandwidth usage since cellular networks are often slower and more expensive than their WiFi and Ethernet counterparts.

However, conducting and reporting measurement-based experiments is not a trivial task as it requires significant experience and knowledge in a wide variety of different areas, such as experimental design, statistics, programming, and scientific writing. As a result, there are several possible questions that may arise during the planning of such an experiment. For example, what are the possible metrics to quantify the characteristic of interest? Where can be found suitable subjects? What instrumentation is available? How to analyze the collected data? What are the expectations with regard to the studies' replicability? We answer these questions by rigorously analysing existing scientific studies on the mobile Web.

To the best of our knowledge, this is the *first* study to provide a well-structured and evidence-based map of the techniques, common empirical practices, and tools used in measurement-based experiments on the mobile Web. Therefore, the **goal** of this paper is to carry out a review of existing studies that conduct measurement-based experiments on the mobile Web. In this study, a total of 640 potentially relevant studies are identified. After a rigorous selection procedure the set of primary studies consists of 28 scientific publications. To perform data extraction, a comparison framework is rigorously defined and applied to the 28 primary studies. Finally, the obtained data is synthesized with the goal of presenting a clear overview of the state of the art on conducting measurement-based experiments on the mobile Web.

The **target audience** for this paper includes both practitioners and researchers that are interested in conducting measurement-based experiments on the mobile Web and that want to be aware of state-of-the-art empirical practices, techniques, and tools used in such experiments.

2 STUDY DESIGN

This study is designed and carried out by following well-accepted methodological guidelines on secondary studies [13, 22, 35]. A complete *replication package* is publicly available[1] for independent replication and verification of our study.

2.1 Goals and Research Questions

The goal of this paper is to identify and classify the characteristics of existing research that conduct measurement-based experiments on the mobile Web from the perspective of both researchers and practitioners. Concretely, this study considers the following research questions:

[RQ1] *What is the state-of-the-art on the **metrics and data management** while conducting measurement-based experiments on the mobile Web?*

[RQ2] *Which **platforms** are considered when conducting measurement-based experiments on the mobile Web?*

[RQ3] *Which **subjects** are used when conducting measurement-based experiments on the mobile Web?*

[RQ4] *What is the state-of-the-art on the **execution** of measurement-based experiments on the mobile Web?*

By answering these research questions we provide an overview of the possibilities, common practices and approaches to carry out measurement-based experiments on the mobile Web. Answering these questions is useful for researchers and practitioners as it can help them with planning and conducting such experiments, while building on a solid foundation coming from the common experience of the community.

2.2 Search and Selection

As shown in figure 1, the search and selection process of this mapping study consists of four main phases. We carried out those phases in a sequential order and independently of the others. In the following we provide the details about each phase.

Initial Search. In this step we perform a query on *Google Scholar*, which is considered to be one of the most comprehensive academic search engines currently available [11]. In addition, it provides unique functionalities that facilitate the practice of snowballing as well as automatically extracting query results from the indexer. The query used to the perform the automated search is given in Listing 1 and is applied to the title of the targeted studies. In essence the search string can be divided into three main components separated by the AND logical operator of which the first one captures the focus on the Web, the second one is about measurement-based experiments, and the third one keeps the focus on the mobile domain. This phase leads to the identification of 232 potentially relevant studies.

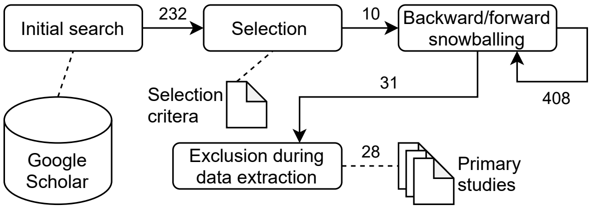


Figure 1: The search and selection process of this study

*("Web" **OR** "browser") **AND** ("Experiment" **OR** "Empirical" **OR** "assessment" **OR** "Analysis" **OR** "Measurement" **OR** "assessing" **OR** "Analysing" **OR** "Measuring") **AND** ("mobile")*

Listing 1: Search string used for the automatic search

Application of Selection Criteria. In this step, we filter the 232 potentially relevant studies by rigorously applying a set of inclusion and exclusion criteria. A study is added to the set of primary studies if it satisfies **all** inclusion criteria and **none** of the exclusion criteria. We used the following inclusion and exclusion criteria:

IC1 – Studies focusing on the mobile Web.
IC2 – Studies reporting measurement-based experiments, i.e., their findings are based on quantitative data collected at run-time (e.g., page load time, energy consumption, etc.).
IC3 – Studies targeting Web apps running either on a smartphone or a tablet.
EC1 – Studies that are not written in English.
EC2 – Studies for which the full text is not available.
EC3 – Secondary or tertiary studies.
EC4 – Studies that are not in the form of a journal article, conference paper, book or book section.
EC5 – Studies that have not been peer reviewed.
EC6 – Studies whose main contribution is not an empirical evaluation.

Each study is manually analysed by applying the adaptive reading depth technique [21], *i.e.,* by incrementally reading the text, starting with the title, abstract, and introduction, and then reading the full text, if necessary.

Furthermore, syntactic duplicates (papers that are exactly the same, *i.e.,* same title, authors, abstract, and venue) are excluded and thus just a single version is kept. If more than one potentially relevant study is about the same experiment (*e.g.,* a conference paper that is extended to a journal version), only one instance is considered; nevertheless, all versions, if they meet the selection criteria, are used during the data extraction phase.

After evaluating all 232 potentially relevant studies, a total of 10 studies met the inclusion and exclusion criteria.

Backward/forward snowballing. The main goal of this phase is to complement the previously described automatic search with a snowballing activity [34]. Snowballing allows us to enlarge the

[1] https://github.com/S2-group/ease-2021-replication-package

Table 1: The comparison framework

Attribute	*Definition*	*Possible Values*
Metrics and data management (RQ1)		
Main aspect	What is the main aspect that the study aims to quantify?	Energy Consumption (EC), Performance (PF), Bandwidth (BW), Caching (C), or Memory Consumption (MC)
Used metrics	What metric is used to report the measurements?	Joules, Page Load Time (PLT), Bytes, Hit Rate
Data analysis	To what extent is the gathered data analyzed?	Descriptive Statistics (DS), Correlation Analysis (CA), Development of Predictive Models (PM), Hypothesis Testing (HT), Effect Size Estimation (ESE)
Replication package	Does the study provide the code and data needed to reproduce the experiments described in the paper?	Instructions, Code & Data (ICD), Code & Data (CD), Code only (C), None (NO)
Platform (RQ2)		
Device type	The device type on which the experiments were run.	Smartphone, Tablet, Emulator
Operating system	The operating system running on the device during the experiment.	Android, iOS, Other
Browser	Which browser is used during the experiments?	Chrome, Safari, FireFox, Modified, Other
Subjects (RQ3)		
Type	Whether the Web apps used in the experiment were real-world industrial projects or demo/toy Websites.	Both, Real, Synthetic
Selection	From which source were the real Web apps selected.	Alexa, No source, List, Other
Hosting	Whether the real Web apps were copied to another server (a mirror) or kept on their own original server.	Original, Mirrored
Number of subjects	Number of Web apps used during the experiment.	Integer
Subjects Provided	Whether the paper explicitly mentions the Web apps used during the experiment (e.g., via their URL).	Yes, No
Experiment execution (RQ4)		
Scope	How the experiment tests the Web apps.	Page load only, Usage scenarios
Focus	Does the experiment focuses on certain Web technologies of the Web apps?	All, HTML, CSS, JS
Tools	What tools (hardware or software) were used to carry out the measurements?	Monsoon power monitor, Google Lighthouse, Custom JavaScript
Network condition	On what network the experiments were run.	WiFi, 3G, 4G, Simulated
Caching	Was browser caching enabled or disabled?	Enabled, Disabled, Not reported

set of potentially relevant studies by (i) considering each study selected in the previous phases and (ii) selecting those papers that are either cited by it (backward snowballing) or citing it (forward snowballing). Note that we performed a closed recursive backward and forward snowballing activity in this study [34]. This leads to 408 additional studies on which we again apply the aforementioned selection criteria. This round lead to the inclusion of 21 additional studies meeting our selection criteria. As a result, a total of 31 primary studies are ready for the data extraction phase.

Exclusion during data extraction. During the data extraction phase it is still possible for papers to be excluded from the set of primary studies. This happens if it turns out that a paper, after reading it full-text, satisfies an exclusion criteria or is found to be a duplicate. For example, we found a semantic triplicate: three papers that had different titles, abstracts and bodies, but actually reported the same experiment [16–18]. In addition, one study was an extension of another study that was already included in our set of primary studies [1, 2]. We merged these papers accordingly, leading to the final set of 28 primary studies.

2.3 Data Extraction

During this phase each primary study is read in detail and the paper's relevant data is extracted by means of a well-structured comparison framework. The comparison framework is presented in Table 1; it is structured around the four research questions of this study. We designed the comparison framework so to facilitate the search for overarching themes and patterns among the primary studies in terms of how they conduct measurement-based experiments on the mobile Web. The comparison framework is initially defined by considering four pilot studies extracted from the 28 primary studies and then it is improved over multiple iterations, while extracting the data for the remaining 24 primary studies.

2.4 Data Synthesis

After filling in the comparison framework for each primary study, we perform a combination of content analysis and narrative synthesis with the goal of gaining insights about measurement-based experiments on the mobile Web. Content analysis allows us to quantify the extracted data, while narrative synthesis refers to the method of synthesizing research in the context of systematic reviews where a textual narrative summary is adopted to explain the characteristics of the primary studies [6, 15].

3 RESULTS

In this section we report the insights gained from our analysis of the extracted data for each research question.

3.1 Metrics and Data Management (RQ1)

Main aspect – Figure 2 shows the frequency of the main aspects across the set of primary studies. As can be observed, the two most frequently considered aspects are performance and energy consumption, respectively. Far less common are studies that examine the impact of bandwidth, cache performance and memory consumption.

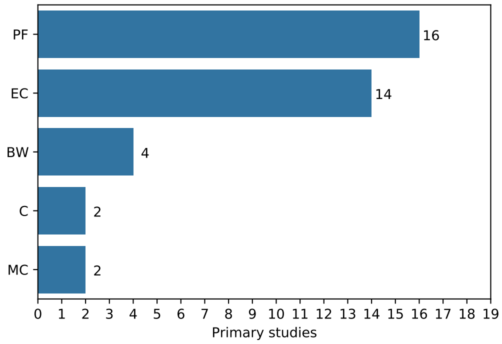


Figure 2: The main considered aspects in the primary studies

Out of the 28 studies there are 10 papers that aim to quantify multiple aspects as their main focus. Not surprisingly, the combination of measuring energy consumption and performance is seen most often followed by studies focusing on energy consumption and bandwidth. They together make up more than half of this category.

Used metrics – The diversity in metrics used to quantify the aforementioned aspects differ greatly. When reporting energy consumption 10 of the 14 studies use Joules to do so. The four papers that deviate from this practice convey their measurements in MAs (S19), mVs (S25) or by comparing them to a baseline (S10, S21)

The four studies measuring bandwidth usage all use bytes or a derivative thereof, e.g. kB (S28, S20, S21, S19).

The cache performance, measured by 2 primary studies, is reported by using the hit rate, defined as the division of saved traffic and total traffic in a visit (S6). On a deeper level more advanced cache performance metrics are used such as the cacheability and actual cache performance of a webpage together with the positive and negative hit and miss ratios (S15).

One of the studies (S1) that report memory consumption uses the Proportional Set Size (PSS) as a metric. PPS is defined as the portion of memory occupied by a process and is composed of the private memory of that process plus the proportion of shared memory with one or more other processes. The other study (S25) that measures memory consumption simply uses MBs.

When it comes to performance we see much more variety in terms of used metrics. A total of 19 different metrics were found. However, this observed influx of different performance metrics is predominantly caused by a single study: S3. Overall, no less than 8 of the 16 studies measuring performance use the Page Load Time (PLT) metric, followed by the SpeedIndex (SI) which is used by 4 studies and time to interactive (TTI), utilized by 2 papers. We have also encountered studies that use a relatively undefined performance metric, such as browser latency (S26) and loading time (S22, S6), but do not give a solid definition and its therefore unclear how and if they actually differ from PLT.

Data analysis – As can be observed in Figure 3 all 28 primary studies analyzed their gathered data using descriptive statistics, i.e. using mean values, standard deviations and presenting plots to get an understanding of the data that has been collected. 10 out of the 28 papers use hypothesis testing to support their findings and decisions with statistical evidence. Effect size estimation, used to measure the strength of the relationship between variables, is utilized by 5 papers. To gain insights into the relationship between variables, 3 studies make use correlation analysis techniques. Finally we found 3 studies that develop prediction models based on their gathered data.

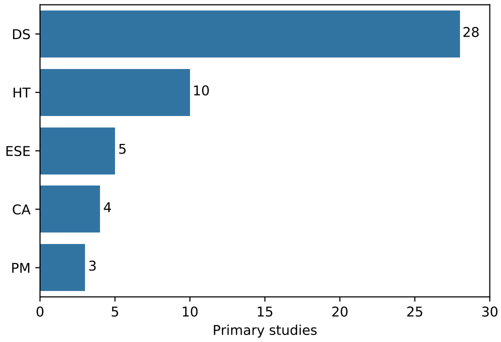


Figure 3: Used data analysis techniques

Replication package – Figure 4 depicts the availability of replication packages across the 28 primary studies. We observe that 20 of the 28 studies do not provide a replication package at all. For the 8 studies that actually do provide a replication package we find that both the contents and the quality of the packages differ significantly. 5 papers equip the reader with a detailed set of written instructions on how to use its contents to replicate the experiment in combination with the the code and the collected data. 2 papers provide the code and data but do not elaborate on how to actually use these (S5, S8) and 1 study only provides the code which consisted of their experimental apparatus (S17).

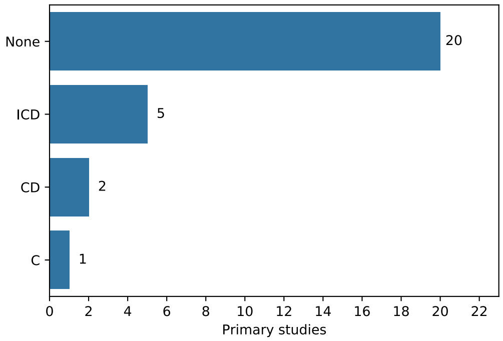
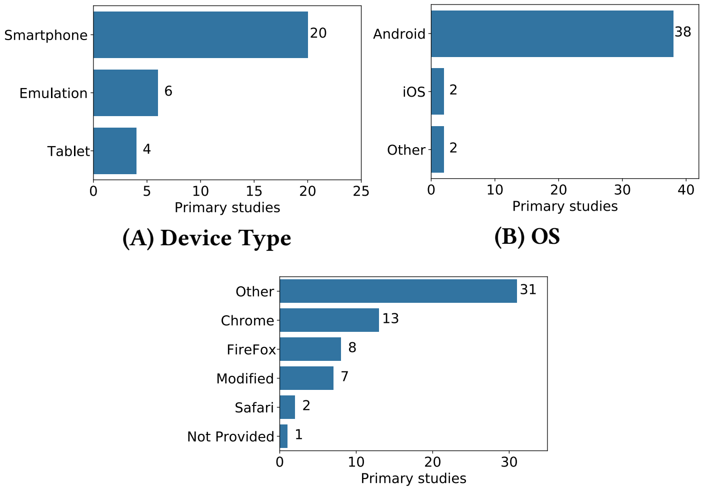


Figure 4: Availability of a replication package

3.2 Platform (RQ2)

Device type – The frequency of device types on which the experiments are carried out is given in Figure 5A. We can see that most studies, 18 of the 28 papers, run their experiments on a smartphone. Most of these smartphones are phones such as these that can be found in the Google Nexus or Samsung Galaxy product line. An interesting exception is a study (S14) that uses an Odroid-XU3 board that contains an Exynos5422 SoC which is also used in the Samsung Galaxy S5 phone. It runs the Android 4 KitKat OS and therefore essentially functions as a proxy for an Android smartphone. The authors do not explicitly motivate their decision for using a single-board computer instead of a real smart device. However, we assume that the Odroid-XU3's built-in power consumption monitoring tool played an important role. The number of papers utilizing a tablet is much less, only 2 of the 28 papers run their experiment on tablet. In addition there are 2 studies that employ both a tablet and smartphone giving us a total of 4 studies running their experiments on a tablet.



(A) Device Type

(B) OS

(C) Browser

Figure 5: Characteristics of the platform

The plot shown in Figure 5A only reports on the number of device types used not on the actual number of devices used. Some of the papers use more than one devices of the same device type to run their experiments on. In total (excluding the papers using both a tablet and a smartphone), 10 of the 28 studies use more than one device to run their experiments on. This is often done to get insights into the effect of different qualities of hardware on the measured results. For example, S13 runs their experiments on both a low budget and high budget flagship smartphone to see how this impacts the measured energy consumption and whether a difference between the two can be found. Similarly, in S5 4 different smartphones are used that were popular in 2015, 2016, 2017, and 2018, each running the most popular Android version in that year ranging from Android 4 to Android 7 which allowed the researchers to do historical studies.

Finally we can see that there are 5 papers that do not carry out their experiments on mobile devices at all, instead they use a form of emulation. Most recent browsers such as Google Chrome and FireFox are equipped with a set of developer tools that allow the user to simulate a range of other devices and browsers to approximate how the page looks and performs on a mobile device. For example, S15 uses a PC running the Chrome browser with its browser's emulation mode activated to make it act as an Android 4.2 native browse so that visited websites return their mobile-version of Web pages. The main reason given for using emulation is because it allows one to leverage the browserâĂŹs programmability, which is not easy on real smartphones and tablets. Another interesting approach is taken by S8 as it makes use of the MONROE platform, which allows them to run measurements with full control of 100 nodes scattered in various locations across four different countries and connected via 11 commercial MBB providers. Each node consists of similar hardware to an average smartphone and is configured to mimic a mobile device browser by setting both the screen resolution and the user-agent accordingly. These studies all emulate a mobile browser. However, S28 emulates an entire operating system as it runs a Windows Phone 6 emulator on a desktop PC and runs a browser within that emulator.

Operating System – The barplot in Figure 5B shows the frequency of operating systems used on the mobile devices (emulation based platforms were excluded) on which the measurement-based experiments were run. In total 42 mobile devices were used over the 28 primary studies. It can be observed that most devices use the Android operating system, namely 38 of the 41. Only 2 devices ran on iOS which were an Apple iPad 2 and an Apple iPhone 4 (S22). In the "other" category we found a device that was equipped with the Maemo 5 OS and one with the Symbian OS (S5).

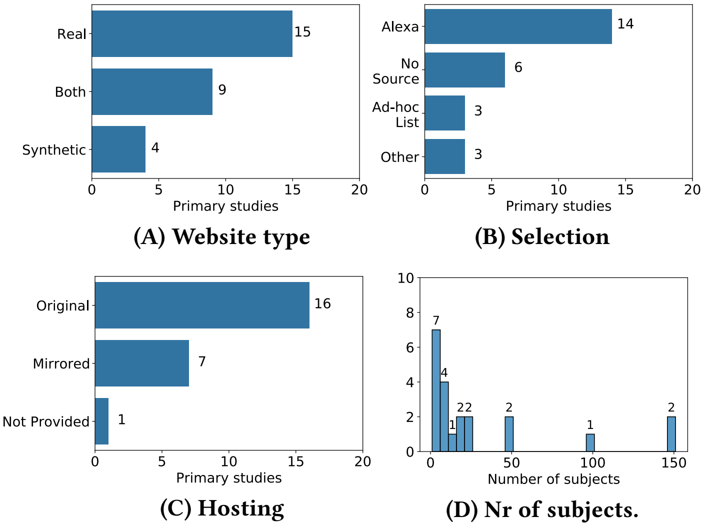
Browser – Looking at Figure 5C we can observe that browsers falling in the category "Other" are the most used browser when doing measurement-based experiments on the mobile web. However, one study is responsible for the large number of browsers in this category, S5, as they tested 4 distinct versions of 6 different browsers falling in the other category. So, 24 of the 31 are the result of this study. Frequently used browsers in the "other" category are the native Android browser and Opera.

After that we see that Google Chrome is the most used browser (13 times) followed by Mozilla FireFox (8). In 6 of the 62 cases a modified browser was used. This was often done to make it easier to measure certain characteristics. For example, in S26 the authors added about 1200 lines of code to 27 files of a WebKit-based browser to make it possible to capture the dependency timeline. Apple's Safari is only used 2 times since only 2 of the 42 devices ran iOS.

Not shown in the plot but interesting nonetheless; for experiments using a form of emulation, Chrome is the most popular browser as its used in 3 of the 8 situations. Firefox is used 2 times and Opera just once

3.3 Subjects (RQ3)

Type – Figure 6A shows the frequency of the types of website considered across the 28 primary studies. It can be observed that 15 of the 28 studies use real websites, i.e. no toy examples, no demo Web apps, no Web apps developed by students or non-professional developers) and no Web apps specifically created for the experiment. However, 13 studies make use of synthetic websites. Studies using both real and synthetic websites often created synthetic copies to see how certain changes on real Web pages impacted the measurement results. For example, S24 first measures the energy consumption of 25 top websites. After that they look at the energy consumption of individual Web elements by copying the Web pages and commenting out specific components to see how this impacts the energy usage. The 4 papers that exclusively used synthetic Web apps created their own Web apps specifically for the experiment. For example, S9 built a Web app that used three different web-based communication protocols (polling, long polling and websockets) to see how they compare in terms of energy consumption. Similarly, S27 measured the energy consumption of a simple Web app that was implemented in 8 different JavaScript frameworks and programmed by computer science students at the master and post-graduate level.



(A) Website type

(B) Selection

(C) Hosting

(D) Nr of subjects.

Figure 6: Characteristics of considered subjects

Selection – The plot shown in Figure 6B shows how the real websites were chosen. Some papers use more than one source, so the number of occurrences does not correspond to the number of primary studies. Noticeable is the high prevalence of Alexa as a source to select websites, 14 of the 28. In 6 cases it was not explicitly mentioned where the website selection is based upon. For example, S19 uses both the New York Times and the Web page of their university but no motivation is given. Three others used another source, specifically: S26 based their website selection on a blog post listing the 10 most visited websites on mobile phone in 2009. S13 and S4 both selected Web apps from a repository of PWAs called PWARocks. In the other category we found one paper that used a set of Javascript benchmarks (S22), a paper for which the website was provided prior to the experiment (S3), and one that scraped Web pages meeting certain requirements (S7).

Hosting – Figure 6C depicts the way real websites (so excluding the two papers that only use synthetic websites) are hosted. It can be observed that most papers prefer to use the original websites' host. However, in 7 of the 24 cases the website(s) were mirrored. This is often done as the researchers want to create a fully controlled environment. They might for example simulate certain network conditions, hosting a website on a server that is in their control makes this easier. One study did not explicitly report how the websites were hosted (S19).

Number of Subjects – The frequency of the number of subjects used in an experiment is depicted in Figure 6D using a histogram with the bin width set to 5. For one study, S11, it was not clear how many subjects were used in total.

We can see that 21 of the 27 studies use 150 subjects or less in their experiment. 7 of these 21 papers use 5 subjects or less. The 6 remaining papers are not included in the graph for the sake of readability as the number of subjects used differs significantly. For example, there are two studies that use 3400 and 95728 subjects respectively (S7, S18). This large number is primarily caused by the fact that analysis was done afterwards. For example, S18 examines 95728 websites by crawling all these websites and downloading all assets loaded by each website, along with a record of all request and response details saved in an HTTP archive record (HAR) file. Then later, all this data is analyzed.

Subjects Provided – In total 14 of the 24 papers that used real websites provide the actual URLs to these websites. This is often done by means of an appendix, a file in the replication package or an in-text table listing all the websites.

3.4 Experiment Execution (RQ4)

Scope – Figure 7 depicts the scope of the experiments. Most studies focus on page load only, namely 21 of the 28. In total 7 papers actually experiment with usage scenarios. For instance S18 simulated a few user interactions to invoke additional content and functionality that initially may be hidden after loading the page to measure bandwidth usage. Three of the 7 papers do both experiments focusing on page load and usage scenarios. In S1 the authors also provide insights into the effect on memory usage when a user scrolls a page and uses multiple tabs in addition to just loading the page.

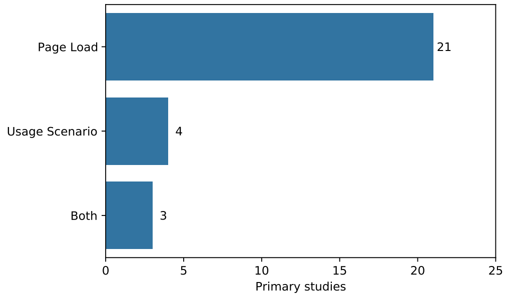


Figure 7: Scope of the experiments in the primary studies

Focus – Out of the 28 primary studies, 24 focus on Web pages as a whole, they do not put a special emphasis on one of the three essential Web technologies (HTML, CSS, JavaScript). The 4 exceptions all focus on JavaScript (S9, S27, S13, S24). For example, S27 implemented a simple Web app in 8 different JavaScript frameworks to see how each framework influences the energy consumption.

Tools – The used instrumentation to do the measurements is of course dependent on what is exactly measured. For the total of 12 papers that measure energy consumption 9 do that by using software based tools. For example, S9 uses two different software based energy profilers: the Trepn profiler and the GreenSpector profiler. The paper states that while hardware based profilers usually offer higher precision, selecting and configuring hardware equipment is complex and can therefore introduce additional bias. Additionally, it requires special equipment which makes reproduction of the experiment more difficult. Another interesting example, S21, feeds tcpdump traces into a radio energy model to get the energy consumption. The other 3 studies use hardware based measurement tools such as the Monsoon power monitor or other multimeters. One paper (S11) argues that while using software based tools results in a simpler and cheaper measurement setting they have several drawbacks. For example, the energy software may be available only for certain mobile devices or Operating Systems, they can cause a form of energy overhead and thus biasing the measurement results or the accuracy of the results strictly depends on the supported power models and the implemented APIs.

For the 13 papers that measure performance its more difficult to pinpoint overarching themes as the tools used are very diverse. When mirroring the original pages some studies inject custom written JavaScript code to measure the PLT (S8, S4). Similarly, S5 uses Boomerang, a JavaScript library that measures performance timings, metrics and characteristics of your user's Web browsing experience[2] when its embedded into the page. Noticeable is the high prevalence of tools created by Google. Two papers (S17, S1) use Telemetry[3], a performance testing framework that allows users to perform arbitrary actions on a set of Web pages (or any android application) and report metrics about it. Two other papers (S3, S2) use Google Lighthouse[4].

To measure bandwidth, tools to inspect network traffic like WireShark are often utilized. Of the two papers that measure memory consumption one created their own app to measure PPS (S6) while another used Google Chrome's Timeline Tool (S25).

Finally we see a lot of custom created tools that manage the orchestrating process. For example, the authors of S26 created an Android application that opens the system's default browser and visit a preprovided list of websites. Other tools frequently used are proxies like Charles and Fiddler and Linux Traffic Control (tc) to simulate network conditions.

Network Conditions – Figure 8 shows the network conditions under which the experiments took place. Again, since some studies experimented with multiple network conditions the number of occurrences does not correspond to the number of primary studies.

It can be observed that most experiments use a real, non-simulated, WiFi network (12 occurrences) followed by a real 3G network (8 occurrences). The real Ethernet connections are primarily because of studies that used a form of emulation, i.e., using a Desktop and mimicking a mobile browser.

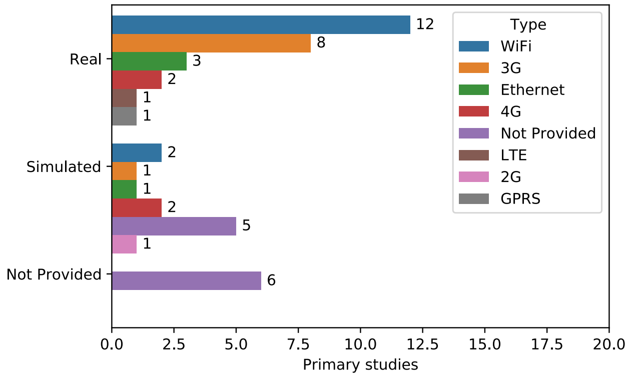


Figure 8: Network conditions while running the experiment

When simulating network conditions, often (5 times) the exact network type is not defined, instead only the down- and upload speeds are given. In 6 cases the paper did not elaborate on the network conditions under which the experiment was carried out.

Caching – We can observe from Figure 9 that most, 12 of the 28, primary studies disabled caching during their experiments to make sure that all requested data will be from the server. This makes sure that for each run of the experiment the same environmental conditions hold. One exception is S22 where the authors conclude that that the loading time required for ten and twenty images is similar after the second run because they didn't clear the cache.

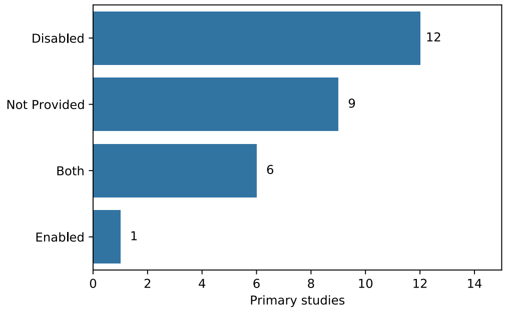


Figure 9: Cache conditions while running the experiment

Six primary studies did experiments with both an enabled and disabled cache. Most of these experiments do this to assess how caching influences the characteristic under measurement. For example, S21 does two types of loads. A cold-cache load where all caches are cleared before loading, and a warm-cache load right after the cold-cache load without clearing any cache. Finally we can see that 9 studies did not give any information about whether their experiments ran with cache enabled or disabled.

4 DISCUSSION

4.1 Main Insights and Recommendations

In this section we discuss the main insights emerging from our results and make recommendations for both researchers and Web developers.

[2] https://github.com/akamai/boomerang
[3] https://chromium.googlesource.com/catapult/+/HEAD/telemetry/README.md
[4] https://developers.google.com/web/tools/lighthouse

Regarding the distribution of considered aspects in the primary studies, we can say that **the focus of research lies primarily on measuring the performance and energy consumption**. This result is not surprising as these two characteristics are very noticeable by users and thus have a great influence on the QoE which in turn, as previously discussed, is vital towards the success of mobile Web apps [12, 23]. However, as a consequence, a relative limited number of measurement-based experiments have been performed on caching, memory consumption, and bandwidth usage. We invite researchers to investigate on those aspects as well since they also contribute towards the perceived QoE of mobile websites.

Below we report our recommendations about the used metrics. We suggest to **use Joules for energy consumption** since it is used in the majority of analyzed primary studies and is widely accepted and understood within the software engineering community. **The landscape of available metrics for performance is highly fragmented**. We presume that this is the result of the introduction of the JavaScript Performance APIs, such as the Performance Timeline[5] and Navigation Timing API[6], in combination with the development of tools like Google Lighthouse, WebpageTest and SiteSpeed.io that allow developers to assess the performance of their Web apps via their own metrics. Page Load Time is still a solid metric for the majority of use cases. However, some papers argue that other metrics such as SpeedIndex and above-the-fold time represent user-perceived load times better [26, 33]. While the choice of a metric depends strongly on the context and goal of the experiment, we advice researchers and practitioners to **clearly define and explain the chosen metrics and, if possible, to explicitly describe how it differs from other popular metrics as well**.

Almost two third of the primary studies report their results without carrying out hypothesis testing. This result is not worrisome *per se* since (i) studies can have an exploratory nature and (ii) the blind application of statistical tests might be problematic as well [5, 20]; however, several primary studies draw strong conclusions and even make recommendations based on their collected data, without providing evidence about whether it is statistically significant or not. Adding to that, only 5 papers do effect size estimation which is often considered to be essential when reporting the results of a statistical analysis [32]. We recommend researchers to **carry out and report in details a proper statistical analysis of the obtained measures, when the goal of the study aims at establishing evidence about a given phenomenon**.

Alarming is the low number of studies providing a replication package. This is unfortunate as replicability is considered to be the major principles of the scientific method and its importance has been been emphasized multiple times over the years [4, 27, 31]. Ideally, scientific results are documented in such a way that their independent verification and replication is fully possible. It may be interesting to look into defining a set of guidelines for replication packages provided by studies doing measurement-based experiments in the context of the empirical software engineering field. On a broader scale there are several existing initiatives like the Open Science movement with its Open Science Framework and rOpenSci which provides a reproducibility guide, including a checklist from which inspiration can be drawn[7] [10].

We applaud the relative high number of studies using more than one device in their experiments; **using more than one device improves the generalizability of the results of the experiment**. While it is generally preferred to use real devices to perform experiments, we understand that the use of emulation can be advantageous in some situations. One of the primary studies mentioned the reason for using a Desktop browser in emulation mode is to make it easier to use the programmability of the browser. However, over the years lot of new tools have been developed to which the required programmability can be possibly delegated as they help streamlining the orchestrating process of setting up and executing measurement-based experiments. A good example of such tools is Android Runner, an extensible framework for automatically executing measurement-based experiments on native and Web apps running on Android devices [19]. Finally we have found that most papers do not explicitly motivate their choice for a certain device or combination of devices. Although we understand that resources are often scarce making the device type a given, we feel that providing a rationale could lead to more insights.

Android is clearly the most used operating system when doing measurement-based experiments on the mobile Web. This corresponds to the latest statistics concerning the worldwide mobile operating system market share as of November 2020 where Android is responsible for 71.18% of the market [30]. Another reason that may have contributed to the high prevalence of the Android operating system is that it is open source and provides a wide software ecosystem. This enables researchers and practitioners to use it as an environment for experiments since it imposes no restrictions on the applications the user can install. This in contrast to for example iOS. However, this does not completely justify the low number of studies doing experiments on devices running iOS. iOS is still covering 28.19% of the mobile operating system market share [30]. In this regard we can argue that the number of studies that use iOS is relatively low. What are the consequences of this in terms of understanding the performance of mobile Web apps on the iOS platform? It is possible that it prevents the optimization of mobile Web apps on the iOS platform since most optimization techniques are based on studies that did their experiments on an Android based device. It might be possible that these findings do not carry over, or not carry over in the same way to the iOS platform, thus resulting in the wrong aspects of Web apps being optimized. We are therefore in favour of doing experiments using iOS based devices with a focus on how the results differ from their Android counterparts. A recent paper investigating the trends and challenges of the mobile software engineering domain mirrors our findings regarding the high amount of scientific contributions featuring only the Android ecosystem [3].

Regarding the Web apps used in the experiments, we see that there is a **strong dominance of the Alexa list** when it comes to a source to select sites from. The Alexa list provides the most popular websites worldwide and thus essentially shows the most visited sites for the average user and in that regard can be considered a solid choice as its representative of the browser behaviour

[5] https://www.w3.org/TR/performance-timeline

[6] https://www.w3.org/TR/navigation-timing

[7] https://ropensci.github.io/reproducibility-guide/sections/checklist

of must users. However, when selecting websites from the top of the list researchers should be aware that **typically the popular top 200 Web pages on Alexa tend to be *highly optimized***. The performance of these Web pages may not be typical and therefore not generalizable. Some papers acknowledge this possible bias and try to mitigate the problem. For example, S5 selects websites from various positions in the list, i.e., 30% from the pages from the bottom of Alea's 1 million websites. Apart from the possible lack of generalizability, there is research that shows that Alexa rankings can be manipulated and change significantly on a daily basis [25]. To combat these shortcomings, the Tranco list has been recently-introduced. The Tranco list is based on the combination of four existing lists (*i.e.,* Alexa, Umbrella, Quantcast, and Majestic). The Tranco list in allows researchers to filter out undesirable (*e.g.,* unavailable or malicious) domains, it is stable over time, and it has been designed for reducing the effort in replicating studies based upon it [24, 25]. We suggest researchers to be aware of the aforementioned initiatives and act accordingly. Finally, **there were a number of primary studies where the Web apps were selected without any proper reasoning and a relative high number of papers that do not provide the actual URLs when using real websites**. This is not desirable as it negatively impacts the replicability of those studies.

The distribution of the scope of the experiments show that most experiments focus on *page load only*, while in practice real users interact with mobile websites through gestures and other interactions. **Experiments based on page load only may not be representative of the actual QoE perceived by the users**, especially if we consider the high amount of JavaScript-based techniques used today for performance improvement, *e.g.,* lazy loading resources.

4.2 Challenges

By examining and analyzing the 'limitations', 'future work' and 'treats to validity' sections of our primary studies we found several overarching themes in terms of challenges encountered when performing measurement-based experiments on the mobile web. More specifically, three main challenges for researchers emerged:

- *low generalizability of obtained results* (16 occurrences). Often this is because of the limited number of hardware devices and websites used;
- *representativeness of the used metrics* (6 occurrences). In most cases this concerns the use of the page load time (PLT) metric. Overall authors find that better metrics are available, but these metrics can possibly influence the measurements or are significantly more difficult to measure;
- *measurement errors* (6 occurrences). This could be caused because of technical limitations of the measurement infrastructure; for example, separating the energy consumed by different processes is a well-known challenge for researchers [28].

The issue of low generalizability is something we have touched upon multiple times above. We suggest that future research should try to employ a more diverse selection of devices and websites to improve the generalizeability of this field of research. Concerning the representativeness of used metrics, the rise of new widely-used tools like Google Lighthouse may improve this situation as we expect that browser vendors will be more and more inclined to cater to developers. Similarly, the issue of possible measurement errors may be mitigated by the development of new tools and modelss, as well as the usage of existing tools such as Android Runner, which supports researchers in following the empirical best practices by design [19].

5 THREATS TO VALIDITY

This section reports on the potential threats to validity of this study. **Internal Validity** – This threat has been mitigated as much as possible by defining and following a strict research protocol, elaborated on in section 2. This research protocol was iteratively defined by discussing it after each iteration among all the co-authors of this study. One limitation may be the possible bias of the used search string. It might not cover all or not return a representative set of papers published in this domain. To alleviate this limitation both backward and forward snowballing was utilized.
External Validity – We already briefly expressed our concerns regarding the small number of papers in the set of primary papers that measure bandwidth, memory consumption and cache performance. They therefore may not be representative of the whole field of research on these characteristics in the context of measurement-based studies on the mobile web. To mitigate this possible threat, we employed an academic search engine as well as used backward-forward snowballing. In addition, the inclusion and exclusion criteria were defined collaboratively among all co-authors of this study. Another potential threat may be the exclusion of papers not written in the English language. However, we deem this treat to be minimal as English is considered to be the main language of science [8]. Finally, the focus on peer-reviewed papers only could be seen to be a risk but is actually intrinsic to our study design since we aim to focus exclusively on the state of the art presented in high-quality scientific studies.
Construct Validity – To mitigate the problem of primary studies not being able to properly answer the chosen research questions we performed an automatic search on all data sources indexed by Google Scholar. This accounts for potential biases due to to publishersĂŹ policies and business concerns. In addition the search string was kept as general as possible to enable a high level of inclusiveness. Moreover the automatic search was complemented by the practice of both forward and backward snowballing. After collecting all relevant studies, we manually carried out the selection process using the chosen inclusion and exclusion criteria as discussed in section II-B2.
Conclusion Validity – Potential biases during the data extraction process were mitigated by discussing the defined classification framework among all co-authors of this study. This way we can guarantee that the data extraction process was aligned with our chosen research questions. Furthermore, we applied well-known best practices on the conduction of secondary studies during each phase of our study [14, 21, 35]. However, possible limitations may be that (i) only one person carried out the search and selection phase and (ii) only one person extracted the actual data from the papers.

6 CONCLUSION

This paper presents a systematic mapping study on conducting measurement-based experiments on the mobile web. In total 640 papers were examined resulting in a set of 28 primary studies which were analyzed via the presented comparison framework to answer our chosen research questions. Below we report the main insights emerging from this study:

(1) The most common aspects considered in measurement-based experiments on the mobile Web are performance and energy consumption. The frequently used metrics to report these measurements are page Load Time, the Speed Index, and Time-to-Interactive for performance and Joules for energy consumption. To measure energy consumption primarily software based tools are used.
(2) All studies analyze their data using descriptive statistics however, only a limited number of papers use hypothesis testing to statistically support their findings.
(3) A limited number of studies provide a replication package.
(4) The most used device type on which to carry out experiments are smartphones running the Android operating system using the Google Chrome browser.
(5) Most experiments use real websites that are selected through the Alexa list and are hosted on their original servers.
(6) Most experiments focus on page load only, with caching disabled, and using a non-simulated WiFi network.

The obtained insights can help practitioners and researchers by providing an evidence-based overview of the state of the art on the common techniques, empirical practices, and tools used to perform measurement-based experiments targeting the mobile Web.

REFERENCES

[1] Inmaculada Ayala, Mercedes Amor, and Lidia Fuentes. 2019. An energy efficiency study of web-based communication in android phones. *Scientific Programming* 2019 (2019).

[2] Inmaculada Ayala, Mercedes Amor, Lidia Fuentes, and Daniel Munoz. 2017. An empirical study of power consumption of web-based communications in mobile phones. In *2017 IEEE 15th Intl Conf on Dependable, Autonomic and Secure Computing, 15th Intl Conf on Pervasive Intelligence and Computing, 3rd Intl Conf on Big Data Intelligence and Computing and Cyber Science and Technology Congress (DASC/PiCom/DataCom/CyberSciTech)*. IEEE, 861–866.

[3] Luciano Baresi, William G Griswold, Grace A Lewis, Marco Autili, Ivano Malavolta, and Christine Julien. 2020. Trends and Challenges for Software Engineering in the Mobile Domain. *IEEE Software* 38, 1 (2020), 88–96.

[4] John E Boylan. 2016. Reproducibility. *IMA Journal of Management Mathematics* 27, 2 (2016), 107–108.

[5] David Colquhoun. 2014. An investigation of the false discovery rate and the misinterpretation of p-values. *Royal Society open science* 1, 3 (2014), 140216.

[6] Daniela S Cruzes and Tore Dybå. 2011. Research synthesis in software engineering: A tertiary study. *Information and Software Technology* 53, 5 (2011), 440–455.

[7] Philippe De Ryck, Lieven Desmet, Frank Piessens, and Martin Johns. 2014. The Browser as a Platform. In *Primer on Client-Side Web Security*. Springer, 25–32.

[8] David G Drubin and Douglas R Kellogg. 2012. English as the universal language of science: opportunities and challenges. *Molecular biology of the cell* 23, 8 (2012), 1399–1399.

[9] Kit Eaton. 2012. How one second could cost amazon $1.6 billion in sales. *Fast Company* 14 (2012).

[10] Erin D Foster and Ariel Deardorff. 2017. Open science framework (OSF). *Journal of the Medical Library Association: JMLA* 105, 2 (2017), 203.

[11] Michael Gusenbauer. 2019. Google Scholar to overshadow them all? Comparing the sizes of 12 academic search engines and bibliographic databases. *Scientometrics* 118, 1 (2019), 177–214.

[12] Jiri Hosek, Michal Ries, Pavel Vajsar, Lubos Nagy, Zdenek Sulc, Petr Hais, and Radek Penizek. 2013. Mobile web QoE study for smartphones. In *2013 IEEE Globecom Workshops (GC Wkshps)*. IEEE, 1157–1161.

[13] B Kitchenham and P Brereton. 2013. A systematic review of systematic review process research in software engineering. (2013).

[14] Barbara Kitchenham and Pearl Brereton. 2013. A systematic review of systematic review process research in software engineering. *Information and software technology* 55, 12 (2013), 2049–2075.

[15] Barbara Kitchenham and Stuart Charters. 2007. *Guidelines for performing Systematic Literature Reviews in Software Engineering*. Technical Report EBSE 2007-001. Keele University and Durham University Joint Report. http://www.dur.ac.uk/ebse/resources/Systematic-reviews-5-8.pdf

[16] Xuanzhe Liu, Yun Ma, Yunxin Liu, Tao Xie, and Gang Huang. 2015. Demystifying the imperfect client-side cache performance of mobile web browsing. *IEEE Transactions on Mobile Computing* 15, 9 (2015), 2206–2220.

[17] Yun Ma, Xuanzhe Liu, Shuhui Zhang, Ruirui Xiang, Yunxin Liu, and Tao Xie. 2015. Measurement and analysis of mobile web cache performance. In *Proceedings of the 24th International Conference on World Wide Web*. 691–701.

[18] Yun Ma, Xuan Lu, Shuhui Zhang, and Xuanzhe Liu. 2014. Characterizing cache usage for mobile web applications. In *Proceedings of the 6th Asia-Pacific Symposium on Internetware on Internetware*. 68–71.

[19] Ivano Malavolta, Eoin Martino Grua, Cheng-Yu Lam, Randy de Vries, Franky Tan, Eric Zielinski, Michael Peters, and Luuk Kaandorp. [n. d.]. A Framework for the Automatic Execution of Measurement-based Experiments on Android Devices. In *35th IEEE/ACM International Conference on Automated Software Engineering Workshops (ASEW-20)*. 61–66.

[20] Blakeley B McShane, David Gal, Andrew Gelman, Christian Robert, and Jennifer L Tackett. 2019. Abandon statistical significance. *The American Statistician* 73, sup1 (2019), 235–245.

[21] Kai Petersen, Sairam Vakkalanka, and Ludwik Kuzniarz. 2015. Guidelines for conducting systematic mapping studies in software engineering: An update. *Information and Software Technology* 64 (2015), 1–18.

[22] K Peterson, S Vakkalanka, and L Kuzniarz. 2015. Guidelines for conducting systematic mapping studies in software engineering: An update. (2015).

[23] Gustavo Pinto and Fernando Castor. 2017. Energy efficiency: a new concern for application software developers. *Commun. ACM* 60, 12 (2017), 68–75.

[24] Victor Le Pochat, Tom van Goethem, and Wouter Joosen. 2018. Rigging Research Results by Manipulating Top Websites Rankings. *CoRR* abs/1806.01156 (2018). arXiv:1806.01156 http://arxiv.org/abs/1806.01156

[25] Victor Le Pochat, Tom Van Goethem, Samaneh Tajalizadehkhoob, Maciej Korczyński, and Wouter Joosen. 2018. Tranco: A research-oriented top sites ranking hardened against manipulation. *arXiv preprint arXiv:1806.01156* (2018).

[26] Ihsan Ayyub Qazi, Zafar Ayyub Qazi, Theophilus A Benson, Ghulam Murtaza, Ehsan Latif, Abdul Manan, and Abrar Tariq. 2020. Mobile web browsing under memory pressure. *ACM SIGCOMM Computer Communication Review* 50, 4 (2020), 35–48.

[27] David B Resnik and Adil E Shamoo. 2017. Reproducibility and research integrity. *Accountability in research* 24, 2 (2017), 116–123.

[28] Rubén Saborido, Venera Venera Arnaoudova, Giovanni Beltrame, Foutse Khomh, and Giuliano Antoniol. 2015. *On the impact of sampling frequency on software energy measurements*. Technical Report. PeerJ PrePrints.

[29] StatCounter. [n. d.]. Desktop vs Mobile vs Tablet Market Share Worldwide Nov 2015 - Nov 2020. ([n. d.]). https://gs.statcounter.com/platform-market-share/desktop-mobile-tablet/worldwide/#monthly-201511-202011 Accessed: 2020-12-15.

[30] StatCounter. [n. d.]. Mobile Operating System Market Share Worldwide Nov 2020. https://gs.statcounter.com/os-market-share/mobile/worldwide. ([n. d.]). Accessed: 2020-12-18.

[31] Victoria Stodden. 2010. The scientific method in practice: Reproducibility in the computational sciences. (2010).

[32] Gail M Sullivan and Richard Feinn. 2012. Using effect size or why the P value is not enough. *Journal of graduate medical education* 4, 3 (2012), 279.

[33] Jamshed Vesuna, Colin Scott, Michael Buettner, Michael Piatek, Arvind Krishnamurthy, and Scott Shenker. 2016. Caching doesn't improve mobile web performance (much). In *2016 USENIX Annual Technical Conference (USENIX ATC 16)*. 159–165.

[34] Claes Wohlin. 2014. Guidelines for snowballing in systematic literature studies and a replication in software engineering. In *Proceedings of the 18th international conference on evaluation and assessment in software engineering*. 1–10.

[35] Claes Wohlin, Per Runeson, Martin Höst, Magnus C Ohlsson, Björn Regnell, and Anders Wesslén. 2012. *Experimentation in software engineering*. Springer Science & Business Media.