Quantifying the Energy Consumption of Android Apps and their Web Counterparts

Anonymous Author(s)

Abstract

Context. Most software vendors now offer native software as well as web-based software on mobile devices. There have been relevant studies for the comparison of the performance and development costs of the two. However, no evidence is still available about how the platform impacts the energy consumption.

Goal. With this study, we aim to empirically assess to what extent the platforms impact the energy consumption of Android devices. In addition, we will also evaluate the impact of the device's distance from the router and the type of network on the energy consumption.

Method. The subjects of our experiment are five apps with more than one billion downloads, and five apps from local vendors with less than one billion downloads. We will perform simple automated operations on these ten apps, and each round of experiments will last two minutes. The independent variable is the app version, and the dependent variable of the experiment is the energy consumption in Joules.

Results. We confirm that the platform has a major effect on energy consumption, and web versions consume more energy than native applications. Correspondingly, the distance and network types have no major effect on energy consumption.

Conclusions. This study provides evidence that using web apps could significantly reduce the battery life of Android devices. In addition, because the distance from the router or the type of network has no main impact on the phone's battery, users can use the applications regardless of their distance from the router or the type of network used. Considering that native apps have a better user experience than web apps, we recommend that users use native apps if possible.

ACM Reference Format:

Anonymous Author(s). 2022. Quantifying the Energy Consumption of Android Apps and their Web Counterparts . In *Proceedings of ACM Conference (Conference'17).* ACM, New York, NY, USA, 11 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 Introduction

Mobile devices are pervasive in modern people's lives as billions of people in the world subscribe to the mobile service. According to a report launched by Global System for Mobile Communications [1], two thirds of the global population had a mobile device, and about 51% population were using mobile internet by the end of 2019 [2]. Having a mobile device, subscribing to a mobile service, and utilizing mobile internet are absolutely regular in our daily lives - even lots of people have more than one mobile device. Moreover, the report claimed that the unique mobile subscribers and mobile internet users will still increase in the following five years. Diverse demand for mobile apps would increase as the market for mobile devices grows. Therefore, the energy consumption problem could be a crucial issue that the app providers and developers must focus on in an attempt to win the mobile apps users' favour in a competitive market.

Due to people's round-the-clock usage of mobile devices, energy consumption has become more crucial than before. For instance, video capturing and using GPS navigation [3] would drain the battery sooner than without using them. On the one hand, several types of usage would shorten the uptime of the mobile device. On the other hand, this would also lower the quality of user experience. Therefore, reducing energy consumption while using mobile apps becomes an imperative problem.

Many companies provide two different experiences to their users for using their apps: the native experience and the web-based experience. However, the energy consumption between these two experiences could be different. Therefore, the motivation of our research is to investigate which apps we daily use could be uninstalled on our mobile devices in order to save our battery volume and improve the quality of user experience. Instead, we could replace those apps with running the application in the browser.

We design and conduct an experiment to compare the difference in energy consumption between web apps and native apps, we would come up with an experiment process. In order to better figure out the difference between them, we use one specific mobile device and one fixed web browser app. Also, several popular apps will be considered in this experiment, e.g. Facebook, Twitter, Amazon, Google Maps, YouTube. We choose these apps not only because people frequently use them but also because several articles claim that they drain the smartphone's battery [4] [5] [6].

The result of our experience could help end users who care about energy efficiency to retain necessary native apps and to uninstall dispensable ones. This kind of end user could be more energy-friendly after uninstalling useless native apps.

2 Related Work

Nowadays, many researchers focus on evaluating the energy consumption of Android devices. Their researches always concentrate on comparing the development approaches [7], different kinds of apps [8], using which kind of wireless communication method [9].

Oliveira *et al.* focus on the different development approaches. Their experiment compares the energy efficiency of commonly used approaches to develop apps on Android: Java, JavaScript, and C/C++. They found that if developers combine these techniques together, the hybrid solution using Java and C++ spent 10x less time and almost 100x less energy than a pure Java solution for one

Conference'17, July 2017, Washington, DC, USA

ACM ISBN 978-x-xxxx-xxxx-x/YY/MM…$15.00
https://doi.org/10.1145/nnnnnnn.nnnnnnn

app. Another app showed that hybrid solution using Java and C++ took 8 percents longer to execute and consumed 11 percents more energy than a hybrid solution using Java and JavaScript [7] Our experiment is about comparing the energy consumption of the same application running the same task on Chrome or a native app in some different situations. But our experiment will not consider the different programming languages used by native apps or web apps. And the download count of apps (large-volume apps or medium-volume apps) will be a block factor in our experiment since we want to investigate the different energy consumption in different block factor groups.

Pathak *et al.* measured energy consumption by using eprof (a fine-grained energy profiler) on two different operating systems (Android and Windows Mobile). Eprof shows surprising finding that more than half of energy in free apps is used in showing advertisements. Then they come up with a case study that reveals that I/O manipulating spends the most of energy. So they propose bundles, a new accounting presentation of app I/O energy, which can significantly reduce the energy consumption [8]. Our experiment does not focus on what type of action uses more energy but wants to figure out if the native app has a lower energy consumption and power consumption than the web app. This can help us choose a way to save energy. We will do the same actions in the same native app or web app to remove the impact of different use cases.

Kalic *et al.* have a hypothesis that different wireless communication technologies have significant energy consumption gaps when transferring the same size of data. They measured the energy consumption of three wireless communication technologies: Bluetooth, WiFi, and 3G. Their result showed that for data transfer of the same size of data, 3G used more energy, and Bluetooth had the best energy efficiency [9]. Our experiment will also use WIFI and 4G as the wireless communication technologies to test which one will consume more energy. Above this, we will take the different platforms (native and web) into consideration to check if these two factors will influence the result. We will not add Bluetooth into our experiment because it is difficult to find a web app that uses this protocol.

Trestian *et al.* wanted to find out if the impact of network-related factors (*e.g.,* network load and signal quality level) will influence the energy consumption when using video-related services. The results tell us that the network load and the signal quality level have a combined significant impact on the energy consumption [10]. Our experiment also cares about if the quality level of our WIFI will result in different energy consumption. However, we add platforms into this experiment to find out if there will be a distinction between different platforms.

Corral *et al.* wanted to investigate if the execution time can approximately estimate the energy consumed of a unit of code. After comparing different experiments' results, they found out that different software benchmarks, data size, and programming languages do not influence the consistency of the ratio between the execution time and the energy consumption [11]. Our experiment has a different scope of energy consumption, and we focus on the whole consumption of native apps or web apps when doing the same task. In our experiment, we will measure energy consumption of the apps in Joules using the batterystats profiler.

Most of the related experiments did not focus on the energy consumption between native apps and web apps. So we want to conduct this experiment to give users guidance on how to save energy and encourage companies to develop energy-efficient applications.

3 Study Design

3.1 Goal and Research Questions

The goal of our study is identified in Table 1, using the template presented by Wohlin *et al.* in [12].

Table 1: Goal Definition

Analyze	Native Apps & Their web versions
For the purpose of	Evaluate the difference
With respect to	Energy consumption
From the point of view of	Software developers & End users
In the context of	Android mobile devices
Result	
Analyze native Apps and their web versions for the purpose of evaluating the difference with respect to their energy consumption from the point of view of software developers and end users in the context of Android mobile devices	

RQ1: *What is the difference in energy consumption between native and web-based versions of mobile apps?*

Based on the download counts in Google Play, we created the following two blocked factors for RQ1.

- Block 1: run the apps that have a download higher than 1 billion (Large-volume apps)
- Block 2: run the apps that have less than 1 billion downloads (Medium-volume apps)

To answer this question we will choose ten applications and their corresponding web versions. Five applications will come from Large-volume apps, and the other will come from Medium-volume apps. We will then design ten general tasks for each app, such as scrolling the main page, sending a message or a comment, clicking into a detail page then clicking back. Finally, we will compare the energy consumption of native and web apps for the Large-volume apps and Medium-volume apps.

Answering this question will help users to make a better choice between native apps or network apps. For some apps, the difference in energy consumption between the two versions of the app is negligible. Therefore, users could choose between native apps or web apps based on their preferences.

RQ2: *How does energy consumption change in different network conditions?*

We think the network signals will affect the energy consumption if the native apps or their web versions are trying to do some network tasks. To investigate the effects of different network signals on energy consumption, we would choose several native apps and web apps to test some network tasks, *e.g.,* uploading 20 photos to the cloud.

We identify the following two sub-questions related to RQ2:

RQ2.1: *How does energy consumption change in different types of network types?*

To answer this question, for a certain network related task, we will compare the power consumption and task execution time between WiFi-connected and 4G-connected devices. By answering this research question, we will be able to check if there are any differences caused by WiFi and 4G signals.

RQ2.2: *How does energy consumption change in different router distance?*

To answer this question, we will set two different physical distances (nearby, 10m) between the Android device and the WiFi router. Then we will check the energy consumption of our devices in these two levels of distance.

3.2 Context and Subjects Selection

Based on four dimensions proposed by Wohlin[13], we perform our experiment covering the following context:

The first dimension is Online versus Offline experimentation. Our research question focuses on the difference between native apps and their web versions, which were downloaded from Google Play or performed on Chrome (for web applications).

The second dimension is Students versus Professionals as subjects. Our goal is to evaluate the energy consumption differences between native apps and their web counterparts. Specifically, we focus on the software energy consumption of native apps and web apps. Therefore, we consider our experiments on the side of software developers.

The third dimension focuses on Toy problems or Real problems. We select real native Apps and web apps to perform our experiments, which are all from a real-world context. And the result of our experiment will show which versions of apps consume less energy, which is also a real-world concerned problem.

The fourth dimension is concerned with Specific experiments versus General experiments. Our experiment is specific because we select certain apps based on the inclusion and exclusion criteria shown in Table 2.

Table 2: Subjects inclusion and exclusion criteria

Inclusion criteria	
I1	Applications that have both web version and native version.
I2	Applications that must require network to work properly (cannot work in offline condition).
I3	Applications that allow the user to scroll up and down so that the application will load new content.
Exclusion criteria	
E1	Applications without one of native and web version.
E2	Applications that are workable without network connection.
E3	Application that are allowed to scroll up and down but with limited content (User will be able to scroll up to the page end easily).

We randomly select ten applications from Google Play that meet selection criteria in Table 2. Five of these apps have a download higher than 1 billion (Large-volume apps), and the remaining five have less than 1 billion downloads (Medium-volume apps).

- Large-volume apps: Twitter, Amazon, YouTube, Facebook, Instagram
- Medium-volume apps: EUShop, Vinted, Marktplaat, Kruidvat, Wish

3.3 Experimental Variables

For our research questions, the independent variable is the App version. We have native apps and their corresponding web apps. The dependent variable for both research questions is the energy consumption of the apps in Joules. Energy consumption (E) is computed via the following formula:

$$E = (\frac{P}{10^6})W \times (\frac{T}{1000})s, \quad (1)$$

where P is the power consumption of the apps in microWatts, and T is the time for each run of the experiment in milliseconds.

3.4 Experimental Hypotheses

In order to answer the two research questions of our study, we have formulated the following hypotheses.

Given that μ_{native} is the average energy consumption of native apps and μ_{web} is the average energy consumption of their corresponding web apps, then the null and alternate hypotheses for Block 1 ($b1$) in RQ1 are defined as:

$$H_0^{b1} : \mu_{native} = \mu_{web}$$

$$H_a^{b1} : \mu_{native} \neq \mu_{web}$$

The null hypothesis states that the average energy consumption of native apps and their corresponding web apps is the same in Large-volume apps. Intuitively, the alternative hypothesis states that there is a significant difference in average energy consumption when using native apps versus their web counterparts in Large-volume apps.

Similarly, given that β_{native} is the average energy consumption of native apps and β_{web} is the average energy consumption of their corresponding web apps, then the null and alternate hypotheses for Block 2 ($b2$) in RQ1 are defined as:

$$H_0^{b2} : \beta_{native} = \beta_{web}$$

$$H_a^{b2} : \beta_{native} \neq \beta_{web}$$

Intuitively, the null hypothesis states that the average energy consumption of native apps and their corresponding web apps is the same in Medium-volume apps. Subsequently, the alternative hypothesis states that there is a significant difference in average energy consumption when using native apps versus their web counterparts in Medium-volume apps.

The first two hypotheses for RQ2.1 follow the same structure of RQ1. For the pl platform, τ_{native} is the average energy consumption of native apps, and τ_{web} is the average energy consumption of their corresponding web apps. And for the network type (nt) factor, β_{WIFI} is the average energy consumption of the apps in WIFI network, and β_{4G} is the average energy consumption of the apps in 4G network. Then the null and alternate hypotheses for RQ2.1 are defined as:

$$H_0^{pl} : \tau_{native} = \tau_{web}$$

$$H_a^{pl} : \tau_{native} \neq \tau_{web}$$

The null and alternate hypotheses for the platform (native/web) state the same as defined in RQ1.

$$H_0^{nt} : \beta_{WIFI} = \beta_{4G}$$

$$H_a^{nt} : \beta_{WIFI} \neq \beta_{4G}$$

The null hypothesis states that there is no difference in energy consumption for any WIFI Network type. Subsequently, the alternative hypothesis states that there is a difference in energy consumption for the WIFI network.

The following two hypotheses are defined for the interaction between platform and network type, where τ_i is the effect of treatment i (Native app/Web app) of the platform factor (pl) and β_j is the effect of treatment j (WIFI/4G) of the network type (nt) factor. Then, the null and alternate hypotheses of RQ2.1 are defined as follows:

$$H_0^{pl,nt} : (\tau\beta)_{ij} = 0 \ \forall i, j$$

$$H_a^{pl,nt} : \exists(i, j) \mid (\tau\beta)_{ij} \neq 0$$

The null hypothesis states that there is no significant difference in the average energy consumption of all possible combinations of the treatments (*i.e.,* platform and network type). The alternative hypothesis states that the average energy consumption of at least one pair of treatments is significantly less than others.

Similarly, the first two hypotheses for RQ2.2 follow the same structure of RQ1. For the platform factor (pl), τ_{native} is the average energy consumption of native apps, and τ_{web} is the average energy consumption of their corresponding web apps. And for the router distance (rd) factor, β_{Close} is the average energy consumption of the apps from nearby router distance, and β_{Far} is the average energy consumption of the apps from far router distance. Then the null and alternate hypotheses for RQ2.2 are defined as:

$$H_0^{pl} : \tau_{native} = \tau_{web}$$

$$H_a^{pl} : \tau_{native} \neq \tau_{web}$$

The null and alternate hypotheses for the platform (native/web) state the same as defined in RQ1.

$$H_0^{rd} : \beta_{Close} = \beta_{Far}$$

$$H_a^{rd} : \beta_{Close} \neq \beta_{Far}$$

The null hypothesis states that there is no difference in energy consumption for any router distance. Subsequently, the alternative hypothesis states that there is a difference in energy consumption for distance from the router.

The following two hypotheses are defined for the interaction between platform and router distance, where τ_i is the effect of treatment i (Native App/Web App) of the platform factor (pl) and β_j is the effect of treatment j (Close/Far) of the router distance (rd) factor. Then, the null and alternate hypotheses of RQ2.2 are defined as follows:

$$H_0^{pl,rd} : (\tau\beta)_{ij} = 0 \ \forall i, j$$

$$H_a^{pl,rd} : \exists(i, j) \mid (\tau\beta)_{ij} \neq 0$$

The null hypothesis states that there is no significant difference in the average energy consumption of all possible combinations of the treatments (*i.e.,* platform and router distance). The alternative hypothesis states that the average energy consumption of at least one pair of treatments is significantly less than others.

3.5 Experiment Design

Based on the subjects, variables, and hypothesis of our experiment, we designed our experiment for RQ1 as a 1 factor - 2 treatments experiment. For answering RQ1, we use a randomized complete design with one factor (*i.e.,* platform), two treatments (*i.e.,* native app and web app), two blocks (*i.e.,* large-volume apps and medium-volume apps), and 5 subjects(*i.e.,* different companies' apps). We repeat every trial of the experiment 20 times.

The specific type of apps (native app and web app) in experiment trials are randomly assigned in order to avoid biased results. Moreover, adopting complete design enables us to explore all possible treatments. The estimated running time for answering RQ1 is $5 \times 2 \times 2 \times 20 \times 2m = 800m = 13.33h$ because 5 subjects, 2 treatments, 2 blocks, 20 repetitions per trial, 2 minutes per run are considered. Table 3 and Table 4 present our design mentioned above for RQ1.

Table 3: Trials for answering RQ1 (Block 1)

	Native app	Web app
Twitter	1st	4th
Amazon	3rd	1st
Youtube	4th	5th
Facebook	2nd	3rd
Instagram	5th	2nd

Table 4: Trials for answering RQ1 (Block 2)

	Native app	Web app
EUShop	2nd	5th
Vinted	5th	4th
Marktplaat	1st	2nd
Kruidvat	3rd	1st
Wish	4th	3rd

It is important to note that, to answer RQ2.1 and RQ2.2, we use the same 5 apps used in RQ1 Block1 (*e.g.,* Twitter, Amazon, YouTube, WhatsApp, Instagram). This is because, by answering RQ2.1 and RQ2.2, we mainly aim to investigate the effects of different network types on energy consumption, and if there are any effects of router distance on energy consumption, respectively. Also, the effects of the two blocking factors (Large-volume apps, Medium-volume apps) on energy consumption are already covered in RQ1.

Moreover, we designed the experiment for RQ2 as 2 factors - 2 treatments (2F-2T) experiment based on the subjects, variables, and hypothesis of our experiment. For the experiment of RQ2.1, we simulate the 4G signal by throttling the WIFI network for the Android device so that the provided bandwidth and bitrate are exactly the same as 4G network. As a result, we restrict the bitrate according to the information from one of the popular carrier companies (Vodafone) [14] in the Netherlands. The upload and download bitrate for simulating 4G networks are set to be 13 Mbps and 36 Mbps respectively. Moreover, for answering RQ2.1, we apply randomized complete design with two factors (*i.e.,* platform and network types), two treatments (*i.e.,* native/web app, and 4G/WIFI), and five subjects(*i.e.,* different companies' apps). The estimated running time for answering RQ2.1 is 13.33h.

For answering RQ2.2, we apply randomized complete design with two factors (*i.e.,* platform and the distance from router), two treatments (*i.e.,* native/web app, and distance), and five subjects(*i.e.,* different companies' apps). The estimated running time for answering RQ2.2 is 13.33h.

3.6 Data Analysis

To answer our research questions, the data obtained from the experiments will be analyzed in four phases: data exploration, check for normality and transformations, hypothesis testing, and effect size estimation.

In the data exploration phase, we will get a preliminary insight about the obtained energy consumption values by (i) inspecting the descriptive statistics of obtained energy measures across all subjects, and (ii) visualizing obtained data using box plots and density plots.

In order to check whether parametric or non-parametric statistical tests can be applied, in the second phase we will analyze the distribution of the measured energy consumption across all subjects. Specifically, we will investigate the normality of its distribution by (i) building Q-Q plots and visually inspecting them, and (ii) applying the Shapiro-Wilk normality test. Moreover, in case the obtained data is not normally distributed, as suggested in [15], we will transform the energy measurement data in order to explore the possibility of having a normal distribution, which can potentially lead to higher statistical power to run parametric statistical tests.

Then, we will conduct statistical tests separately in each research question in the phase of hypothesis testing. All statistical tests will be performed with $\alpha = 0.05$ as significance level. If the data comes from a normal distribution, we will apply the paired t-test to fit the 1-factor-2-treatments experiment design on RQ1. However, if the assumptions of the paired t-test are not met, Wilcoxon Rank-Sum test will be applied for the non-parametric statistical test. For RQ2.1 and RQ2.2, we will apply a Two-Way ANOVA statistical test since there are two factors in each experimental design. If the assumptions of the Two-Way ANOVA are not met, Scheirer–Ray–Hare test [16] or Aligned Rank Transform ANOVA [17] will be applied.

Finally, the magnitude of differences among our treatments will be estimated by Cliff's Delta effect size measure.

3.7 Replication Package

The replication package of the experiment is publicly available [18]. It contains the subjects of the experiment, the Python source code we developed for orchestrating the experiment, its raw data, and the R scripts for the data analysis.

4 Experiment Execution

This section provides a detailed description of the infrastructure we set up for the experiment and the software and hardware devices we used.

Figure 1 shows an overview of the infrastructure of our experiment. The infrastructure of the experiment consists of two devices: (i) an Android device on which the subjects of the experiments will

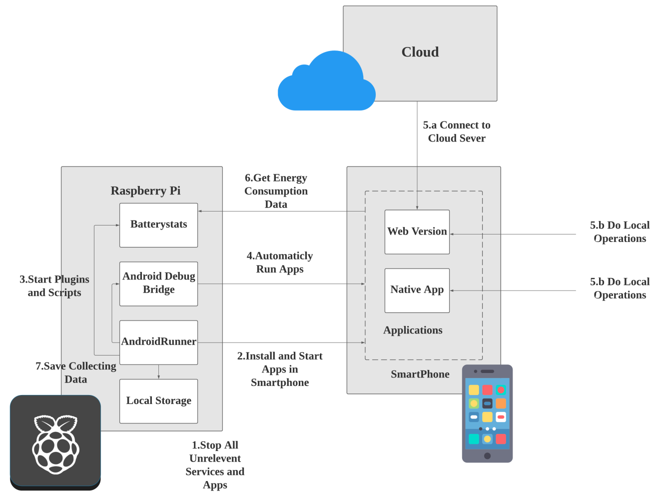

Figure 1: Overview of the infrastructure

be running, and (ii) a Raspberry Pi, which will be used to orchestrate the experiment and is responsible for collecting experimental data from the Android device.

For orchestrating the execution of all the runs of the experiment we make use of Android Runner [19]. Android Runner is a Python framework for automatically executing measurement-based experiments involving both native and web applications running on Android devices. In Android Runner, at first experiments are defined in the form of a JSON file. Then, the entire execution of the experiment is managed by Android Runner with a combination of Python scripts and Android Debug Bridge (ADB) commands. With the help of Batterystats plugin, Android Runner can directly measure the power consumption of Android devices. Batterystats is a software-based profiler for estimating the power consumption of mobile devices. Although the measurements that are based on hardware are more accurate, they requires expensive external equipment, and for batterystats, the estimation error is within 5% in 95% of the analyzed methods [20]. Therefore in our experiment, we will make use of the batterystats profiler.

For orchestrating the experiment, Android Runner will run on a Raspberry Pi 4B and the subjects will be running on a real mobile device. The technical specifications of the Raspberry Pi and the mobile phone are listed in Table 5.

Table 5: Technical Specifications for devices

	Device	Mobile Phone
Manufacturer	Raspberry Pi	Samsung
Model	Raspberry Pi 4B	Galaxy J7 Duo
Operating system	Debian	Android 8.0.0
CPU	4x1.5GHz Cortex-A72	2x2.2 GHz Cortex-A73
Memory	4 GB RAM	32GB 3GB RAM

Both the Raspberry Pi and the Android device run under the same network. To eliminate the influence of the network environment on the experiment, the Raspberry Pi and the Android device are the only devices connected to the router. Furthermore, during our experiment, we disable the charging mode of the Android device to ensure that we can obtain accurate experimental data.

Before running the experiment, we obtain the power_profile.xml [21] file of the mobile device using APKTool [22], this file provides the electric current intensity for the component and CPU states thus enable batterystats to estimate the power consumption. As shown in Figure 1, there are seven steps involved in each round of our experiment:

- Close all third-party apps and disable services such as location service, Bluetooth and push notifications service.
- Install and run the subjects of the experiment through Android Runner
- Start custom plugins(*i.e.,* batterystats and ADB)
- Send a custom command through ADB so that the apps can automatically run customised tasks.
- The apps are either connected to the web server or do local operations to complete the tasks.
- After two minutes of running and profiling, the power consumption data is collected by batterystats.
- Store the obtained data in local storage.

5 Results

5.1 Data Exploration

To better understand the data, we conduct data exploration in this part. The energy consumption across all 10 apps and 10 web-based apps ranges between 92.99 Joules and 272.08 Joules, with a mean of 146.25 Joules.

Table 6: Subjects of the study

App	Platform	Min	Median	Mean	Max
Amazon	Native	142.67	148.86	152.23	162.12
Amazon	Web	156.79	164.48	163.56	170.63
Facebook	Native	98.6	112.97	111.55	123.66
Facebook	Web	118.21	146.11	144.65	182.93
Instagram	Native	97.27	111.13	127.59	172.44
Instagram	Web	159.87	166.79	166.56	173.7
Twitter	Native	118.94	120.56	121.02	124.39
Twitter	Web	142.81	197.53	201.96	272.08
Youtube	Native	190.13	193.07	192.7	196.02
Youtube	Web	141.11	147.34	147.48	156.79
Kruidvat	Native	94.87	96.53	96.84	99.86
Kruidvat	Web	139.83	148.74	147.85	154.91
Marktplaats	Native	103.04	104.78	105.18	112.57
Marktplaats	Web	223.45	228.94	229.14	237.16
Myeushop	Native	99.71	101.07	101.27	104.71
Myeushop	Web	113.51	118.99	118.73	125.3
Vinted	Native	104.71	107.35	107.37	109.55
Vinted	Web	113.92	120.02	119.58	125.71
Wish	Native	98.5	103.12	103.48	107.73
Wish	Web	141.2	148.06	148.33	154.91

Table 6 shows a breakdown of the energy consumption of each application. The table shows that the Kruidvat consumes the least energy among the native applications, and the Marktplaats consumes the most energy among the web applications. On top of this, most web applications consume more energy than their native version. Except for Youtube, its native application consumes more energy than its web version.

Figure 2, Figure 3 and Figure 5 give us a deeper insight into the relationship among energy consumption, platform, distance from the router, and network type. The box plots in Figure 2 and Figure 3 depict the relationship between energy consumption and platform. We observe that whether we consider the block factor in RQ1, the application consumes less energy on the native platform than their web version. Also, we find the same phenomenon in the density plot in Figure 2 and Figure 4.

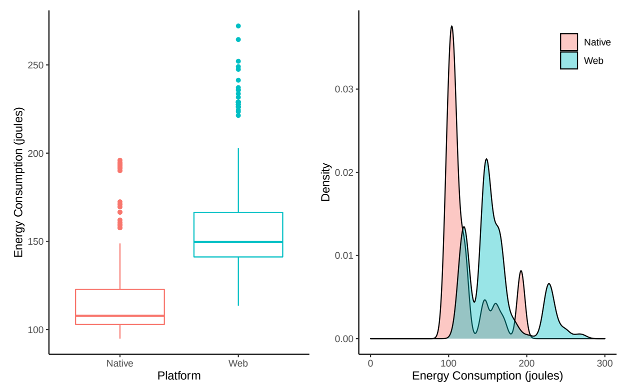

Figure 2: Energy consumption and density plot for each treatment in RQ1

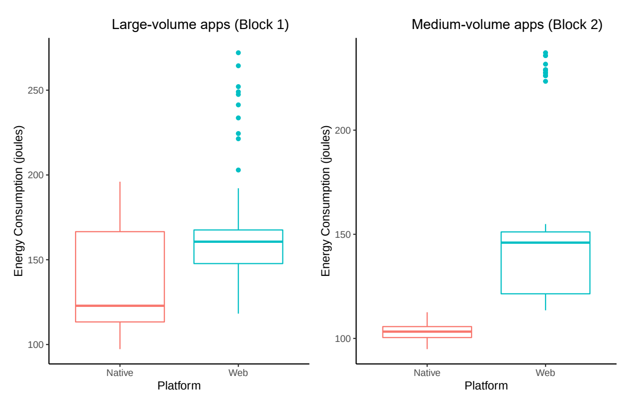

Figure 3: Energy Consumption for each treatment and each block in RQ1

The left part of Figure 5 depicts the relationship between energy consumption and two factors, platform and type of network. We observe that the different network types have less impact on the energy consumption than the platform. Moreover, when the device uses Wi-Fi, the energy consumed by the applications on the native platform is less than the one on the web version.

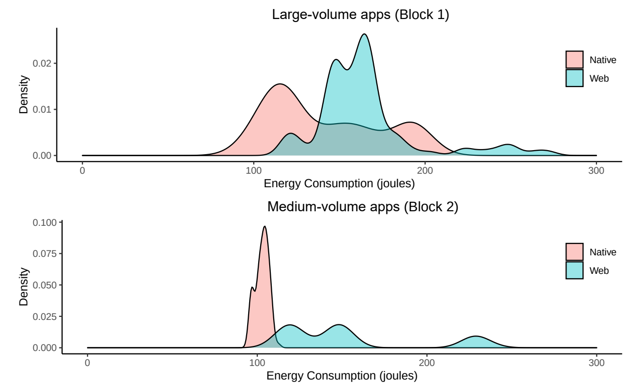

Figure 4: Density for each treatment and each block in RQ1

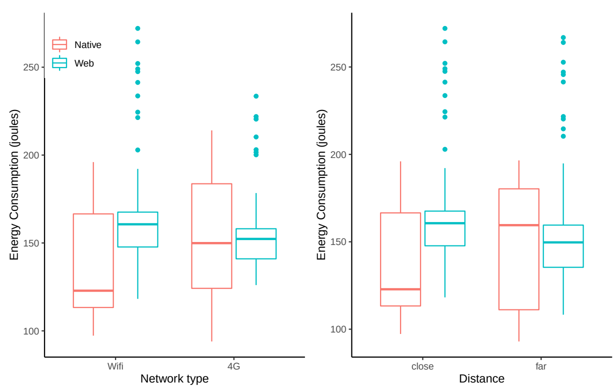

Figure 5: Energy Consumption for each treatment in RQ2.1 (left) and RQ2.2 (right)

The right part of Figure 5 shows the relationship between energy consumption and two factors. One factor is the distance between the Android device and the router. The other one is the platform. We observe that the different distances have less impact on the energy consumption than the platform. It is counter-intuitive that the application's energy consumption for web applications will become less when the device is far away from the router.

In Figure 6, we show the density plots for each treatment in RQ2.1 and RQ2.2. Since we have observed the difference of energy consumption between platforms, we now focus on the one between the other factors which are network type and router distance. In the first two subplots of Figure 6, we observe that there are large overlap areas between different network types. Also, in the last two subplots of Figure 6, there are large overlap areas between different router distances.

5.2 Check for normality and transformations

Before conducting statistical tests, we perform normality tests on our data. We can only perform parametric statistical tests if the data conform to the normal distribution. Otherwise, we can only perform non-parametric statistical tests [23].

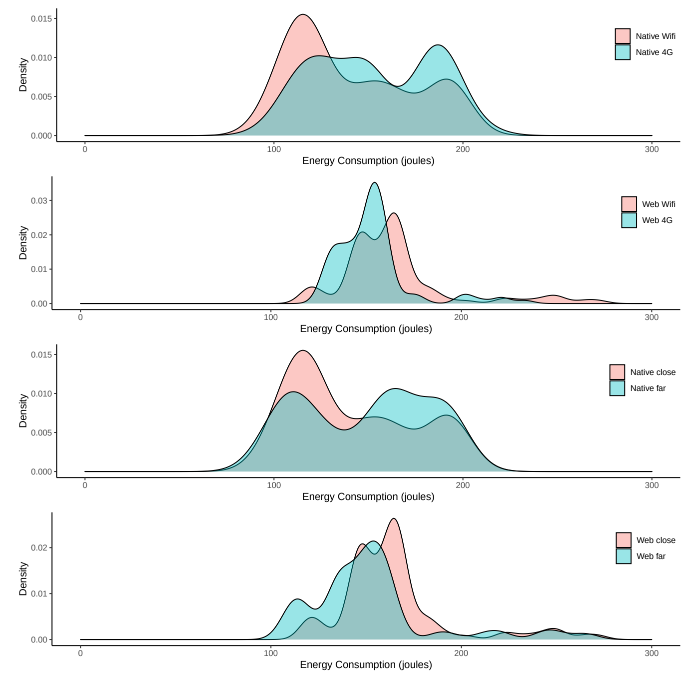

Figure 6: Density plots for each treatment in RQ2.1 and RQ2.2

Visually, these data do not conform to a normal distribution. We performed Q-Q plots of our data against random samples from a normal distribution to prove our conjecture. If our data and random data conform to the same distribution, the points on the Q-Q plot will form a line that is roughly straight. Figure 7 shows no data set conforming to a normal distribution. Furthermore, we performed the Shapiro-Wilk normality test. The results of the Shapiro-Wilk test presented in Table 7 show that no data set has a p-value greater than 0.05. Therefore, we reject the null hypothesis that the data are from a normally distributed population.

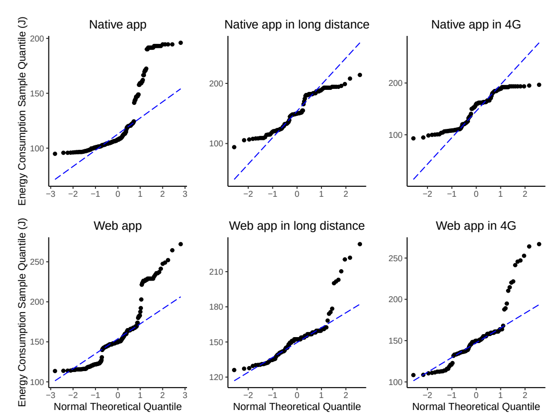

Figure 7: Q-Q plot for energy consumption

However, it is often beneficial to have normal distributed data since it allows us to utilize parametric statistical tests. Therefore,

Table 7: Shapiro-Wilk test for the energy consumption

Data Set	p-value
Native apps	2.2×10^{-16}
Web apps	2.24×10^{-12}
Native apps in long distance	1.79×10^{-6}
Web apps in long distance	1.84×10^{-9}
Native apps in 4G signal	4.28×10^{-5}
Web apps in 4G signal	4.60×10^{-10}

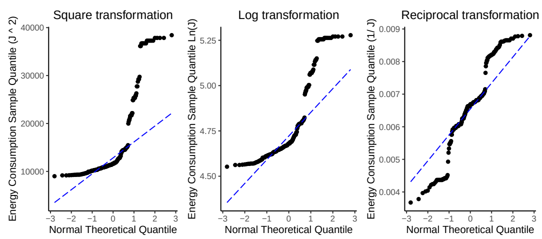

Figure 8: Q-Q plot for 3 transformations in Native app

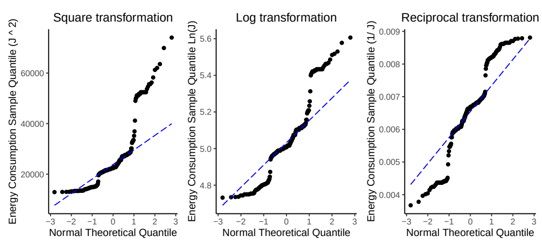

Figure 9: Q-Q plot for 3 transformations in Web app

besides examining normality on original data, we also do the checking after transforming energy measurement data. The squared, nature log, and reciprocal transformation are applied in the data transformation procedure.

Figure 8 and Figure 9 present our normality checking for the first two data sets after three transformations. We observe that none of the transformations provide us with any indication of normality. All six data sets were transformed by three mentioned metrics, followed by checking of Q-Q plot and performing Shapiro-Wilk normality test. However, all transformed data sets reject the null hypothesis that the data is normally distributed.

To conclude, none of our measured energy consumption data sets is normally distributed, even after conducting three different transformations.

5.3 Hypothesis testing

In this section, we will conduct related tests on the hypotheses corresponding to different research questions.

5.3.1 Research Question 1 As the hypothesis that our data comes from a normal distribution is rejected, we perform the Wilcoxon rank-sum test [24] based on our experiment design. The p-value produced by the Wilcoxon rank-sum test for RQ1 is 2.2×10^{-16}. Therefore, we reject the null hypothesis that the energy consumption means of native apps and their web versions are the same. This result demonstrates that applications on different platforms have different energy consumption.

We then focus on two blocks separately, *i.e.,* Block1 (Large-volume apps) and Block2 (Medium-volume apps), and conduct a Wilcoxon rank-sum test for each block. The test results are shown in the Table 8.

Table 8: Wilcoxon rank-sum test for RQ1

	p-value
Block 1 (Large-volume apps)	7.7×10^{-7}
Block 2 (Medium-volume apps)	2.2×10^{-16}

The table shows that both blocks reject the null hypothesis that the energy consumption means of native apps and their web versions are the same. Therefore, we conclude that there are differences in the energy consumption of the native app and the web app in either block.

5.3.2 Research Question 2.1 In this research question, we apply the Scheirer–Ray–Hare test [25] since it is a non-parametric test used for a two-way factorial design. The results of p-value we derive from this test are presented in Table 9. According to the result in the Table 9, we can reject the null hypothesis of H_0^{pl} claiming that the energy consumption of native app is the same as the one of web app. Also, we can reject the null hypothesis of $H_0^{pl,nt}$ which states that there is no interaction effect between platform (native app/ web app) and wireless technology on energy consumption. However, we cannot reject the null hypothesis of H_0^{nt} stating that there is no difference in energy consumption for any wireless technology.

To conclude, this test result allows us to demonstrate that (i) using the application in different platforms has an impact on energy consumption, (ii) the platform and the wireless technology interact in affecting the energy consumption.

Table 9: Scheirer–Ray–Hare test for RQ2.1

Factor	p-value
Platform	1.2×10^{-4}
Network Type	7.9×10^{-1}
Platform & Network Type	8.0×10^{-5}

5.3.3 Research Question 2.2 The Scheirer–Ray–Hare test is conducted in this research question due to its property of non-parametric for two-way factorial design. Table 10 shows the results of the test. According to the results, we can reject the null hypothesis of H_0^{pl} which claims the same statement as RQ2.1. Moreover, we can reject the null hypothesis of $H_0^{pl,rd}$ which claims that there is no interaction effect between platform and router distance on energy consumption. Nevertheless, we cannot reject the null hypothesis of H_0^{rd} which states that there is no difference in energy consumption for any router distance.

In conclusion, the test result provides us with evidence that (i) using the application in different platforms has an impact on energy consumption (the same as RQ2.1), (ii) the platform and the router distance interact in affecting the energy consumption.

Table 10: Scheirer–Ray–Hare test for RQ2.2

Factor	p-value
Platform	7.3×10^{-4}
Distance	3.68×10^{-1}
Platform & Distance	1.5×10^{-4}

To further support our results of RQ2 derived by Scheirer–Ray–Hare test, we applied another hypothesis test - Aligned Rank Transform (ART) ANOVA [26]. The results are shown in Table 11 and Table 12. The magnitude of all p-values produced by ART ANOVA are the same as the one obtained by Scheirer–Ray–Hare test. Therefore, we derive the same conclusion as Scheirer–Ray–Hare test.

Table 11: Aligned Rank Transform ANOVA for RQ2.1

Factor	p-value
Platform	2.2×10^{-4}
Network Type	3.6×10^{-1}
Platform & Network Type	1.0×10^{-5}

Table 12: Aligned Rank Transform ANOVA for RQ2.2

Factor	p-value
Platform	4.3×10^{-4}
Distance	6.42×10^{-1}
Platform & Distance	1.6×10^{-4}

5.4 Effect size estimation

In this section, we focus on the effect size of the result of our Wilcoxon rank-sum test in RQ1. The Cliff's delta measure is applied since it is a non-parametric measure that is suitable for our study. When we consider the amount of difference between two different platforms (native apps and web apps) in RQ1, a large effect size (0.65) is found. When we further consider Block1 and Block2, a medium effect size (0.40) and a large effect size (1.0) are found.

Our effect size measure result can be supported by the density plots in Figure 2 and Figure 4. The density plot for RQ1 shows a large difference between native app and web app. For Block1, there are some overlap areas between native app and web app while the plot still presents differences between two groups. For Block2, there is no overlap between two groups, therefore giving us the result representing a large effect size.

6 Discussion

From the results of the statistical tests obtained in the previous section, we can now elaborate on our research questions. We first elaborate on two blocks in RQ1 separately. For Block 1, the results of the Wilcoxon rank-sum test shows that p-value is less than 0.05 which means that we can reject the null hypothesis. Therefore, we can say that there is a statistically significant difference in energy consumption between the native apps and their web counterparts. For Block 2, from the results of the Wilcoxon rank-sum test, we can also accept the alternative hypothesis that there is a statistically significant difference in energy consumption between native apps and their corresponding web apps. From Figure 3, we can also infer that the mean energy consumption of native apps is less than their web version in both two blocks. The density plot for RQ1 shown in Figure 4 also reveals that web apps consume more energy than native apps. We conjecture that these results are mainly due to the download-parse-execute process that the browser engine does when executing the Web app, whereas native apps run mostly locally and perform network requests only for obtaining data from the Cloud.

For RQ2.1, we use Scheirer–Ray–Hare Test to investigate energy consumption differences between different platforms, and further take the network type into consideration. The results imply that the platform has main effects on energy consumption while the network type has no effect on energy consumption. Looking at the trend of density plots shown in Figure 6, we can also infer that there is no significant difference between Wifi and 4G. Moreover, the test results also reveal that platform and network type do have interaction effects on energy consumption.

For RQ2.2, we take two factors (router distance and platforms) into consideration, and apply the Scheirer–Ray–Hare Test to investigate the energy consumption difference between these two factors. For effect between different platforms, we derive the same results as in RQ2.1. However, the test result for the router distance factor has a p-value bigger than 0.05 which claims that there is no main effect on energy consumption. Moreover, the test results also show that there is an interaction effect between platform and router distance on energy consumption. The patterns between far and close shown in the density plot are also similar.

Finally, we apply the Cliff's delta effect size measure to evaluate the magnitude of differences between two different platforms. This yields a large effect size between native apps and their web counterparts. Besides, the test results also reveal that Medium-volume apps blocks have a larger difference than Large-volume apps blocks.

According to our results, the energy consumption of their native app is less than their web version. Therefore, depending on the characteristics of web apps that they are developing, developers can apply different techniques to make their web apps more energy-efficient. For example, developers may use less JavaScript code, server side caching, or build static web apps, where applicable.

According to our empirical results, router distance and network types show interaction effects with type of apps on energy consumption, which can be useful for developers to consider in reducing energy consumption of their apps. If the developers need a precise analysis of the native or web apps with respect to the router distance or network type, they can run experiments to test if far distance and 4G signals will increase the energy consumption. Based on the type of apps, developers can reduce the energy consumption by lessening the data flow between client and server or use offline-first/caching solutions, such as service workers for web apps [27, 28].

As for users, it is convenient to use native apps rather than their web versions because almost every native app consumes less energy than their web version. Compared to native apps with 4G network, native apps with Wifi network tend to consume less energy. Finally, if users have the choice to use their phone near or far from the router, it is beneficial to use native apps close to the router.

7 Threats To Validity

We have analyzed the validity of our experiment based on four types of classifications, which were defined by Cook and Campbell [29]. We will describe these four types of classification of threats to validity of our experiment in the following sections:

7.1 Internal Validity

7.1.1 Maturation Maturation can be a threat to validity in our repetitions of tests. The current run might be affected by previous runs. In order to reduce the effect, we set the Android runner to take intervals of 10 seconds. And each task will take exactly 2 minutes' execution time. Further, for web applications, we cleared up the browser cache after each run; for Android applications, we also killed its background process after each run. In this way, we made sure that the current run will not be affected by previous runs in our repeated trials.

7.1.2 Reliability of measures Reliability of measures can be a threat to internal validity because some factors of our device may affect this. For example, in our case, we want to collect the energy consumption data of applications, but other functions of our device will also consume some energy. To mitigate this, before our experiment, we set the brightness of the screen to minimum value, we killed all unnecessary background processes and we kept the phone in the same place.

7.2 External Validity

7.2.1 Interaction of selection and treatment One threat to the external validity is that the population of our subjects may not represent the generalization related to our results. In our experiment, the selection of Android applications and web applications was not totally random. We randomly selected 10 applications from Google Play that meet the requirements described in section 3.3. Five of these applications have a download higher than 1 billion (Large-volume apps), and the remaining five have less than 1 billion downloads (Medium-volume apps). The random selection process reduced the possibility of introducing biases. However, the filter process may affect the external validity and result that our results may not be generalized. We still applied the filter process for the reason that it can help specify the scope of our selected experiment.

7.2.2 Interaction of setting and treatment The interaction of setting and treatment may be a threat to external validity in the situation that the experiment scene can not or is difficult to reappear in the real world. In our experiment, we used Samsung Galaxy J7 Duo with Android 8.0.0 as the operating system, which is commonly used in the real world. And we used Chrome to do the experiment with web applications, which is also the most commonly used browser in the real world. Therefore, our experiment was performed in a realistic environment.

7.3 Construct Validity

7.3.1 Definition of constructs We used the GQM method to define the constructs of our experiment in our experiment design stage, which was 2 weeks before we performed the experiment. We derived research questions and metrics related to our experiment goal from our GQM tree. Using the GQM method, we mitigated the inadequate pre-operational explanation of constructs.

7.3.2 Mono-operation bias Our experiment has only one independent variable which makes it possible to unable to represent the theory. However, for each research question, we use 10 subjects and perform 20 repetitions on one single factor. Also, we use different treatments corresponding to research questions. In this way, we mitigated the mono-operation bias.

7.4 Conclusion Validity

7.4.1 Low statistical power The threat deals with the situation that there might be a significant difference but the statistical test does not reveal it due to the low number of data points. Since we had three parts of tests and due to the time constraint, for each test, we did 20 trials to mitigate this. However, we can increase the number of trials in our future research to make sure our results are statistically significant.

7.4.2 Violated assumptions of statistical tests This threat deals with violated assumptions of result data before doing the statistical analysis. To mitigate this, we performed a normality test on our data before performing further statistical tests.

8 Conclusions

In this paper we designed and conducted an experiment to investigate if the energy consumption is different between applications and their web version with two other factors (network types and router distance). Based on our experimental results, we can conclude that the platform has the main effect on energy consumption. Also, for both blocks in RQ1, web versions of the apps consume more energy than their native versions. However, the other two factors (network types and router distance) for RQ2 have no significant effects on energy consumption, but have interaction effects with the platform. These findings will help developers improve the user experience and decrease energy consumption by updating the apps of their web versions. For users, they can choose to use native apps over web versions to reduce energy consumption and thus keep their smartphone's battery healthy.

As for future work, our experiment can be extended by using more treatments in the factor of router distance. This will help researchers figure out whether a longer distance between the router and android device can affect energy consumption. Another way to extend our research is to use more than 20 apps in the experiment so that it allows us to collect more data to explore energy consumption differences between the native apps and their web versions. If the dataset is large enough, it may allow us to conduct parametric tests which can produce more powerful results.

References

[1] "Representing the worldwide mobile communications industry," www.gsma.com, accessed: 2022-01-17.

[2] G. Intelligence, "The mobile economy 2020 gsma," *Intelligence*, 2020.

[3] Z. Zhuang, K.-H. Kim, and J. P. Singh, "Improving energy efficiency of location sensing on smartphones," in *Proceedings of the 8th international conference on Mobile systems, applications, and services*, 2010, pp. 315–330.

[4] "The most battery-draining apps of 2020," www.techrepublic.com/article/the-most-battery-draining-apps-of-2020/, accessed: 2022-01-17.

[5] "Facebook app is killing your phone's battery life," https://www.cbsnews.com/news/facebook-app-is-killing-your-phones-battery-life/, accessed: 2022-01-17.

[6] "Android battery killers: 10 worst apps that drain phone battery," https://www.makeuseof.com/tag/android-battery-killers-drain-worst-apps/, accessed: 2022-01-17.

[7] W. Oliveira, R. Oliveira, and F. Castor, "A study on the energy consumption of android app development approaches," in *2017 IEEE/ACM 14th International Conference on Mining Software Repositories (MSR)*, 2017, pp. 42–52.

[8] A. Pathak, Y. C. Hu, and M. Zhang, "Where is the energy spent inside my app? fine grained energy accounting on smartphones with eprof," in *Proceedings of the 7th ACM European Conference on Computer Systems*, ser. EuroSys '12. New York, NY, USA: Association for Computing Machinery, 2012, p. 29–42. [Online]. Available: https://doi.org/10.1145/2168836.2168841

[9] G. Kalic, I. Bojic, and M. Kusek, "Energy consumption in android phones when using wireless communication technologies," in *2012 Proceedings of the 35th International Convention MIPRO*, 2012, pp. 754–759.

[10] R. Trestian, A.-N. Moldovan, O. Ormond, and G.-M. Muntean, "Energy consumption analysis of video streaming to android mobile devices," in *2012 IEEE Network Operations and Management Symposium*, 2012, pp. 444–452.

[11] L. Corral, A. B. Georgiev, A. Sillitti, and G. Succi, "Can execution time describe accurately the energy consumption of mobile apps? an experiment in android," in *Proceedings of the 3rd International Workshop on Green and Sustainable Software*, ser. GREENS 2014. New York, NY, USA: Association for Computing Machinery, 2014, p. 31–37. [Online]. Available: https://doi.org/10.1145/2593743.2593748

[12] C. Wohlin, P. Runeson, M. Höst, M. C. Ohlsson, B. Regnell, and A. Wesslén, *Experimentation in software engineering*. Springer Science & Business Media, 2012.

[13] C. Wohlin, P. Runeson, M. Höst, M. Ohlsson, B. Regnell, and A. Wesslén, *Experimentation in Software Engineering - An Introduction*. Kluwer Academic Publishers, 2012.

[14] "State of mobile networks: Netherlands (september 2018)," www.opensignal.com/reports/2018/09/netherlands/state-of-the-mobile-network, accessed: 2022-01-17.

[15] S. Vegas, "Analyzing software engineering experiments: Everything you always wanted to know but were afraid to ask," in *Proceedings of the 40th International Conference on Software Engineering: Companion Proceeedings*, ser. ICSE '18. New York, NY, USA: Association for Computing Machinery, 2018, p. 534–535. [Online]. Available: https://doi.org/10.1145/3183440.3183466

[16] R. R. Sokal, "The principles and practice of statistics in biological research," *Biometry*, pp. 451–554, 1995.

[17] L. A. Elkin, M. Kay, J. J. Higgins, and J. O. Wobbrock, "An aligned rank transform procedure for multifactor contrast tests," *arXiv preprint arXiv:2102.11824*, 2021.

[18] Anonymous authors, "Replication package of the study," https://anonymous.4open.science/r/android-apps-web-counterparts, accessed: 2022-01-24.

[19] I. Malavolta, E. M. Grua, C.-Y. Lam, R. de Vries, F. Tan, E. Zielinski, M. Peters, and L. Kaandorp, "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)*. ACM, 2020.

[20] D. Di Nucci, F. Palomba, A. Prota, A. Panichella, A. Zaidman, and A. De Lucia, "Software-based energy profiling of android apps: Simple, efficient and reliable?" in *2017 IEEE 24th international conference on software analysis, evolution and reengineering (SANER)*. IEEE, 2017, pp. 103–114.

[21] "Measuring power values," https://source.android.com/devices/tech/power/values, accessed: 2022-01-17.

[22] "Apktool," https://github.com/iBotPeaches/Apktool, accessed: 2022-01-17.

[23] M. R. Harwell, "Choosing between parametric and nonparametric tests," *Journal of Counseling & Development*, vol. 67, no. 1, pp. 35–38, 1988.

[24] W. J. Conover, *Practical nonparametric statistics*. john wiley & sons, 1999, vol. 350.

[25] C. J. Scheirer, W. S. Ray, and N. Hare, "The analysis of ranked data derived from completely randomized factorial designs," *Biometrics*, pp. 429–434, 1976.

[26] H. Luepsen, "The aligned rank transform and discrete variables: A warning," *Communications in Statistics-Simulation and Computation*, vol. 46, no. 9, pp. 6923–6936, 2017.

[27] I. Malavolta, "Beyond native apps: web technologies to the rescue!(keynote)," in *Proceedings of the 1st International Workshop on Mobile Development*, 2016, pp. 1–2.

[28] I. Malavolta, K. Chinnappan, L. Jasmontas, S. Gupta, and K. A. K. Soltany, "Evaluating the impact of caching on the energy consumption and performance of progressive web apps," in *Proceedings of the IEEE/ACM 7th International Conference on Mobile Software Engineering and Systems*, 2020, pp. 109–119.

[29] T. Cook and D. Campbell, "Quasi-experimentation: Design and analysis issues for field settings 1979 boston," *MA Houghton Mifflin.*