The Energy Consumption Difference between Native App and Their Web Version

Chaoran Li
2688458
VU Amsterdam
c11.li@student.vu.nl

Chang Liu
2686797
VU Amsterdam
c8.liu@student.vu.nl

Chih-Chieh Lin
2700266
VU Amsterdam
c2.lin@student.vu.nl

Shengyu Lu
2697748
VU Amsterdam
s3.lu@student.vu.nl

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[1]. However, no evidence is still available about how 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 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:

Chaoran Li, Chang Liu, Chih-Chieh Lin, and Shengyu Lu. 2020. The Energy Consumption Difference between Native App and Their Web Version . In *Green Lab 2020/2021 - Vrije Universiteit Amsterdam, September–October, 2020, Amsterdam (The Netherlands).* ACM, New York, NY, USA, 11 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

Green Lab 2020/2021, September–October, 2020, Amsterdam, The Netherlands

ACM ISBN 978-x-xxxx-xxxx-x/YY/MM…$15.00
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 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 app would increase as growth of market of mobile device. 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 experience 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,

[1] www.gsma.com

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.[2,3,4]

The result of our experience could help end users who care about energy efficiency to retain necessary native app and to uninstall dispensable ones. This kind of end users 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 [4], different kinds of apps [5], using which kind of wireless communication method [6].

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 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 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 [4] 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 company size (small or big) will be a block factor in our experiment, 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 advertisement. 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 [5]. 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 gap when transferring 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 [6]. 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 [7]. 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 not influence the consistency of the ratio between the execution time and the energy consumption [8]. 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. We will measure energy consumption by using batterystats, and it will retrieve joule that we use in one experiment.

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-efficiency applications.

3 EXPERIMENT DEFINITION

3.1 Goal

The goal of our study is identified in Table 1, using the template presented by Wohlin et al. in [9].

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	

3.2 Questions

3.2.1 *Research Question 1.* What is the difference in energy consumption between native and web-based versions of mobile apps?

Large companies often have more funds and a more standardized development cycle, and they have better control over their products. Based on this reason, we create two blocked factors:

- Block 1: run the apps from companies that have large volumes.
- Block 2: run the apps from companies that have small volumes.

To answer this question, we will choose ten applications and their corresponding web versions. Five applications will come from companies with large volumes, and the other will come from small

[2] www.techrepublic.com/article/the-most-battery-draining-apps-of-2020/
[3] www.cbsnews.com/news/facebook-app-is-killing-your-phones-battery-life/
[4] www.makeuseof.com/tag/android-battery-killers-drain-worst-apps/

companies in the start-up phase. We will then design ten general tasks for each app, such as scrolling the main page, sending a message or a comment and clicking into a detail page then clicking back. Finally, we will compare the difference in energy consumption inside each application to determine if the company with more funds and time has better production control.

Answering this question will help users to make a better choice between native apps or network apps. For example, in apps from some companies, the difference in energy consumption between the two versions is negligible. Therefore, users could choose the native apps or web apps based on their preference. If the energy consumption difference between the two versions in one company is significant, people are more inclined to choose the energy-friendly version when using their product.

3.2.2 Research Question 2. 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 show the effects of different network signals on energy consumption, we would choose several certain native apps and web apps to test some network tasks: like uploading 20 photos to the cloud.

We identify the following two sub-questions related to this question:

- Research Question 2.1. How does power consumption change in different types of network types?

To answer this question, we will set a contrast experiment. For a certain network related task, we will compare the power consumption and task execution time between WiFi-connected and 4G-connected devices. We want to check if there were any differences caused by WiFi and 4G signals.

- Research Question 2.2. How does power consumption change in different router distance?

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

3.3 Metrics

To answer the aforementioned questions quantitatively, we use metrics below to help us measure the experiment results.

- Power Consumption: We will measure this value in microwatts. This value will be used to compute energy consumption.
- Energy Consumption: It is calculated by the following formula:

$$\text{Energy} = (\frac{\text{Power}}{10^6})W \times (\frac{\text{Time}}{1000})s$$

This is our indicator for evaluating the difference in energy consumption between the native apps and web apps.

3.4 GQM-Tree

Figure 1 shows the visual representation of the GQM-Tree.

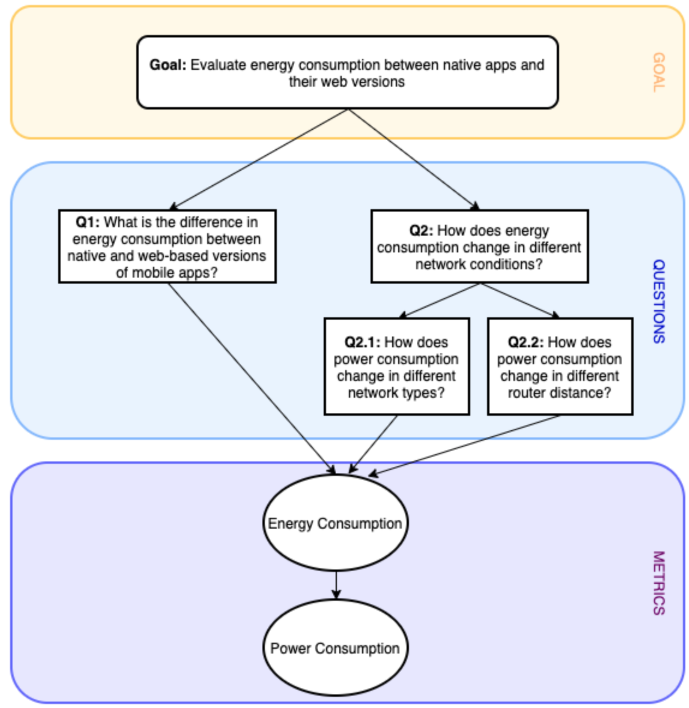

Figure 1: GQM-Tree

4 EXPERIMENT PLANNING

4.1 Subjects Selection

Based on four dimensions proposed by Wohlin[10], 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), so it is impossible for us to influence the inner characteristics of these applications.

The second dimension is Students versus Professionals as subjects. Since we are studying the energy consumption of native Apps and web Apps, which is part of these apps' soft requirements, we consider our experiments more 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 which meet the following requirements:

- have web versions;
- developed by large companies (for RQ1);
- developed by small companies(for RQ1);
- have network-required functions(for RQ2);
- can scroll up and down on one certain page;
- be popular.

We random select ten applications from Google Play that meet the above requirements. Five of these apps have a download higher than 1 billion (large companies), and the remaining five have less than 1 billion downloads (small companies).

- Apps developed by big companies: Twitter, Amazon, YouTube, Facebook, Instagram
- Apps developed by small companies: EUShop, Vinted, Marktplaat, Kruidvat, Wish

4.2 Experimental Variables

For our research questions, our independent variable is the App version. We have native apps and their corresponding web apps. The dependent variable of our study is the energy consumption of apps in Joules. The value of energy consumption will be derived by *Batterystats* [11]. Moreover, we will utilize the value of energy consumption to do the hypothesis test.

4.3 Experimental Hypotheses

To conduct our experiment, three groups of hypothesis are formulated to test our assumptions. Our experiment evaluates the energy consumption between the native apps and their web versions.

For research question 1 Block 1, we test the null hypothesis that the energy consumption means of native apps and their web versions are same in big company block:

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

versus the alternative hypothesis that the energy consumption mean of native apps and their web versions are not the same in big company block

$$H_1 : \mu_{native} \neq \mu_{web}$$

For research question 1 Block 2, we test the null hypothesis that the energy consumption means of native apps and their web versions are the same in small company block:

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

versus the alternative hypothesis that the energy consumption mean of native apps and their web versions are not the same in small company block

$$H_1 : \mu_{native} \neq \mu_{web}$$

For research question 2.1, we have some definitions of i, j and τ, β list below:

τ_i: effect of treatment i (Native App/Web Version) of factor A (Platform)

β_j: effect of treatment j (WIFI/4G) of factor B (Network Type)

$(\tau\beta)_{ij}$: effect of the interaction between τ and β

and we test the null hypothesis that

$$H_{0A} : \tau_{Native} = \tau_{Web} = 0$$
$$H_{0B} : \beta_{WIFI} = \beta_{4G} = 0$$
$$H_{0AB} : (\tau\beta)_{ij} = 0 \quad \forall \text{i,j}$$

versus the alternative hypothesis that

$$H_{1A} : \exists i | \tau_i \neq 0$$
$$H_{1B} : \exists j | \beta_j \neq 0$$
$$H_{1AB} : \exists\,(i, j) \mid (\tau\beta)_{i,j} \neq 0$$

For research question 2.2, we have some definitions of i, j and τ, β list below:

τ_i: effect of treatment i (Native App/Web Version) of factor A (Platform)

β_j: effect of treatment j (Close/Far) of factor B (Router Distance)

$(\tau\beta)_{ij}$: effect of the interaction between τ and β

and we test the null hypothesis that

$$H_{0A} : \tau_{Native} = \tau_{Web} = 0$$
$$H_{0B} : \beta_{Close} = \beta_{Far} = 0$$
$$H_{0AB} : (\tau\beta)_{ij} = 0 \quad \forall \text{i,j}$$

versus the alternative hypothesis that

$$H_{1A} : \exists i | \tau_i \neq 0$$
$$H_{1B} : \exists j | \beta_j \neq 0$$
$$H_{1AB} : \exists\,(i, j) \mid (\tau\beta)_{i,j} \neq 0$$

4.4 Experiment Design

We design this study separately due to different conditions of research questions. For answering RQ1, we apply randomized complete design with one factor (*i.e.* type of apps), two treatments (*i.e.* native app and web app), two blocks (*i.e.* top companies and start-up companies), and 5 subjects(*i.e.* different companies' apps). We repeat every trial of the experiment 20 times. The specific types of app (native app and web app) in experiment trials are randomly assigned in order to avoid biased results. Moreover, adopting complete design enable 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 2 and Table 3 present our design mentioned above for RQ1.

Table 2: 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 3: 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

5 apps used in RQ1 Block1 - Twitter, Amazon, YouTube, WhatsApp, Instagram - are also utilized in our experiments of RQ2.1 and RQ2.2.

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)[5] in the Netherlands. The upload and download bitrate for simulating 4G network are set to be 13Mbps and 36Mbps respectively. Moreover, for answering RQ2.1, we apply randomized complete design with two factors (*i.e.* type of apps 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 13.33*h*.

For answering RQ2.2, we apply randomized complete design with two factors (*i.e.* type of apps 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 13.33*h*.

4.5 Data Analysis

Three main phases will be conducted in our data analysis work: data exploration, check for normality and transformations, hypothesis testing, and effect size estimation.

Our aim in the data exploration phase is to obtain preliminary insight into our experiment results. The summary statistics of energy consumption measure is computed, followed by visualization using histograms, box plots, scatter plots and density plots.

In order to check if we can apply parametric statistical test in our energy measurement data, normality test and Q-Q plots are conducted. Data transformation is applied in order to obtain more options for having normally distributed data allowing us have better statistical power.

We conduct statistical tests separately in each research question in the phase of hypothesis testing. All of the statistical tests are conducted with $\alpha = 0.05$. For RQ1, we plan to apply paired t-test in an attempt to fit the 1-factor-2-treatments experiment design. If the assumptions of the paired t-test are not met, Wilcoxon Rank-Sum test would be applied for the non-parametric statistical test. For RQ2.1 and RQ2.2, we plan to apply 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 or Aligned Rank Transform ANOVA would be applied.

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

5 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 2 shows two main components in our experiment. The first one is the Raspberry Pi. The Raspberry Pi a low cost, credit-card sized computer which integrates the CPU and GPU in a single integrated circuit, with the RAM, USB ports, and other components soldered onto the board for an all-in-one package. Raspberry Pi will run the software that coordinates all execution, and it is responsible for storing experimental data. The second one is an Android device that runs the subjects of the experiment.

[5] www.opensignal.com/reports/2018/09/netherlands/state-of-the-mobile-network

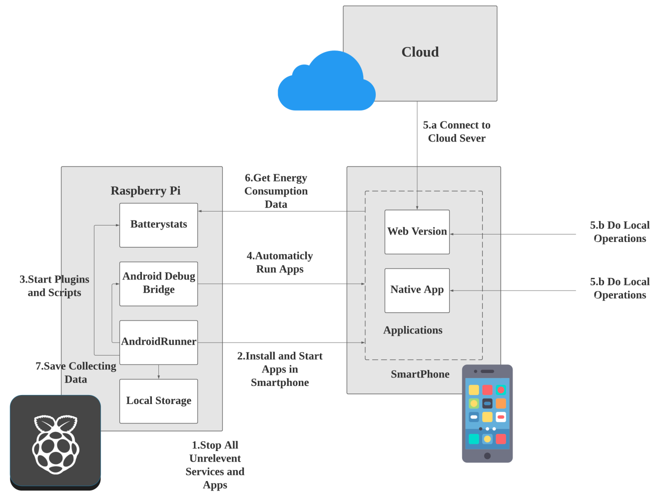

Figure 2: Overview of the infrastructure

Table 4: 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

We will use Android Runner [12] to coordinate the execution of the entire experiment. Android Runner is a frameworks that could automating measurement-based experiments. Furthermore, it is written in python and entirely open source. So it can efficiently work with plugins provided by other users. This characteristic is essential for us to be able to execute experiments smoothly. textcolorblueFirstly, it can work with Android Debug Bridge(ADB), which allows us to communicate with Android devices. The ADB command facilitates a variety of device actions, such as scrolling the application page. We can use ADB to automate executing such commands to complete our experiment. Secondly, Android Runner can be used to measure Android device's power consumption by directly calling batterystats. Batterystats estimates the power consumption of the mobile phone based on software. 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 [13]. Therefore, we will choose batterystats for our experiments.

In our experiments, the Android Runner will run on a Raspberry Pi, and the device on which the subjects are running is an Android mobile phone. The Raspberry Pi and the mobile phone's technical specifications are shown in Table 4.

To eliminate the influence of the network environment on the experiment as much as possible, the mobile phone and the Raspberry Pi will run under the same network. Also, they are the only two devices connected to router. In the experiment, we will turn off the charging mode of the Android device to ensure that we can obtain accurate experimental data.

Before experimenting, we will obtain the battery information file of the mobile phone through apktool, this file provides the electric current intensity for the component and CPU states thus enable batterystate to estimate power consumption states. As shown in Figure, there are seven steps in each round of experiments.

- Close all the software and disable any services such as location service, Bluetooth and push notifications service.
- Install and run the corresponding software through the Android runner
- Start custom plugins(i.e. batterystats and ADB)
- ADB will send custom command so that the apps can automatically running customize tasks.
- The apps are either connected to the webserver or do local operations to complete the tasks.
- After two minutes of running and profiling, the power consumption data is collected by batterystats.
- The data then is stored in local storage.

6 RESULTS

In response to our research questions, we conduct corresponding experiments and collect the following six data sets.

- Energy consumption for native applications
- Energy consumption for web applications
- Energy consumption for native applications in long distance
- Energy consumption for web applications in long distance
- Energy consumption for native applications in 4G signal
- Energy consumption for web applications in 4G signal

6.1 Data Exploration

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

Table 7 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 3, Figure 4 and Figure 6 give us a deeper insight into the relationship among energy consumption, platform, distance from the router, and network type. The box plots in Figure 3 and Figure 4 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 3 and Figure 5.

The left part of Figure 6 depicts the relationship between energy consumption and two factors, platform and type of network. We observe that the different network type 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.

Table 5: Subjects of the study

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

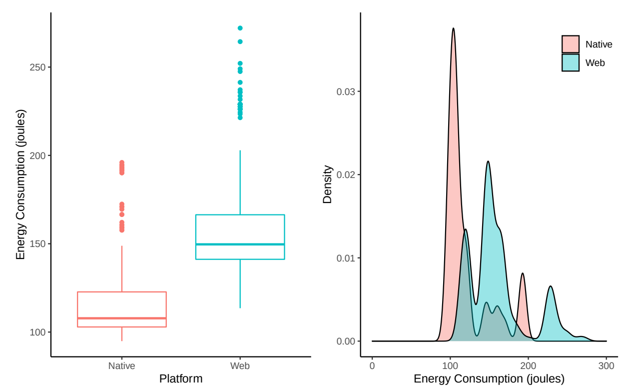

Figure 3: Energy consumption and density plot for each treatment in RQ1

The right part of Figure 6 shows the relationship between energy consumption and two factors, distance from the Android device to the router and 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 application will become less when the device is far away from the router.

In Figure 7, we show the density plots for each treatments 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 7, we observe that there are large overlap areas between different network types. Also, in the last two

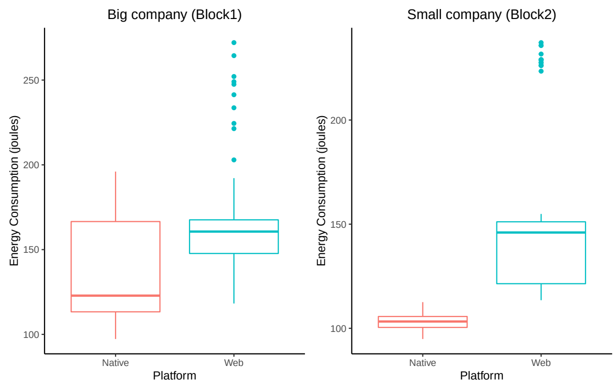

Figure 4: Energy Consumption for each treatment and each block in RQ1

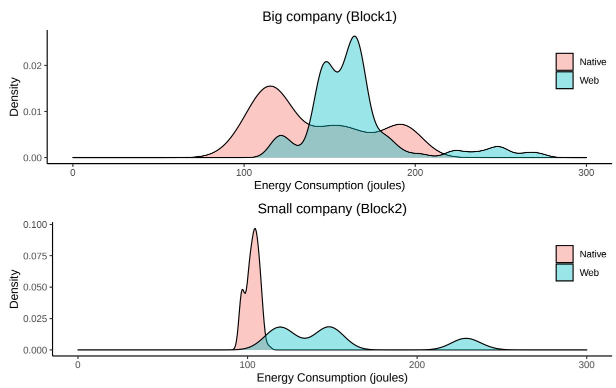

Figure 5: Density for each treatment and each block in RQ1

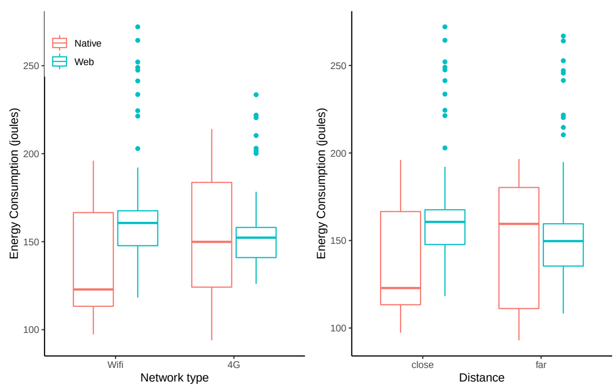

Figure 6: Energy Consumption for each treatment in RQ2.1 (left) and RQ2.2 (right)

subplots of Figure 7, there are large overlap areas between different router distances.

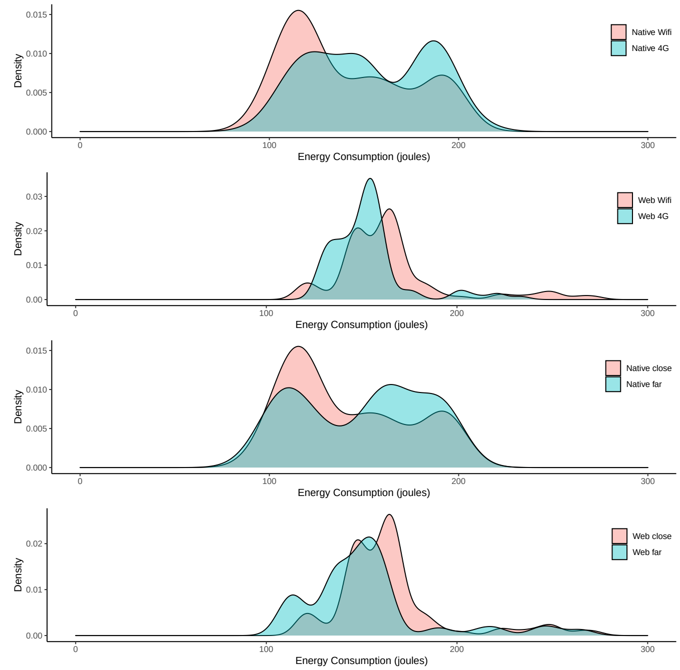

Figure 7: Density plots for each treatment in RQ2.1 and RQ2.2

Table 6: Shapiro-Wilk test for 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}

6.2 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 [14].

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 8 shows no data set conforming to a normal distribution. Furthermore, we performed the Shapiro-Wilk normality test. The results of Shapiro-Wilk test presented in Table 6 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.

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

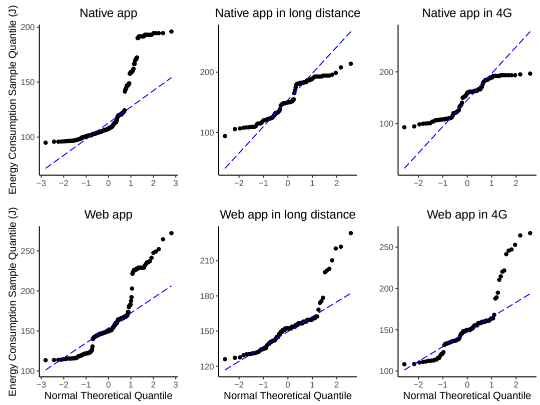

Figure 8: Q-Q plot for energy consumption

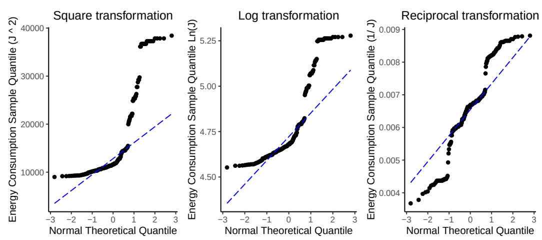

Figure 9: Q-Q plot for 3 transformations in Native app

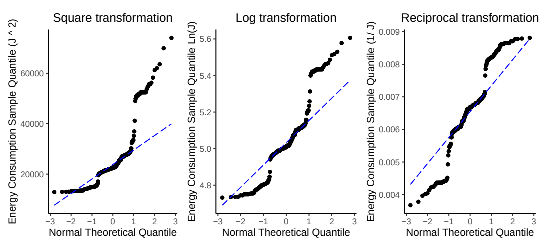

Figure 10: Q-Q plot for 3 transformations in Web app

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 9 and Figure 10 present our normality checking for first two data sets after three transformations. We observe that none of transformation provide us with any indication of normality. All of six data sets were transformed by three mentioned metrics, followed by checking of Q-Q plot and performing Shapiro-Wilk normality test. However, all of transformed data sets reject the null hypothesis that the data is normally distributed.

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

6.3 Hypothesis testing

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

6.3.1 Research Question 1. As the hypothesis that our data comes from a normal distribution is rejected, we perform Wilcoxon rank-sum test [15] based on our experiment design. The p-value produced by 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 (big company) and Block2 (small company), and conduct a Wilcoxon rank-sum test for each block. The test results are shown in the Table 7.

Table 7: Wilcoxon rank-sum test for RQ1

	p-value
Block 1 (big company)	7.7×10^{-7}
Block 2 (small company)	2.2×10^{-16}

The table shows that both blocks rejected 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.

6.3.2 Research Question 2.1. In this research question, we apply Scheirer–Ray–Hare test [16] since it is a non-parametric test used for a two-way factorial design. The result of p-value we derive from this test are presented in Table 8. According to the result in the Table 8, we can reject the null hypothesis of $H_{0A}^{RQ2.1}$ claiming that energy consumption of native app is the same as the one of web app. Also, we can reject the null hypothesis of $H_{0AB}^{RQ2.1}$ 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 $H_{0B}^{RQ2.1}$ stating that there is no difference in energy consumption for any wireless technology.

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

Table 8: 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}

6.3.3 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 9 shows the results of the test. According to the results, we can reject the null hypothesis of $H_{0A}^{RQ2.2}$

which claims the same statement as $H_{0A}^{RQ2.1}$. Moreover, we can reject the null hypothesis of $H_{0AB}^{RQ2.2}$ which claims that there is no interaction effect between platform and router distance on energy consumption. Nevertheless, we cannot reject the null hypothesis of $H_{0B}^{RQ2.2}$ which states that there is no difference in energy consumption for any router distance.

In conclusion, the test result provide us with evidence that (i) using the application in different platform 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 9: 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}

6.3.4 Another hypothesis test for RQ2. To further support our results of RQ2 derived by Scheirer–Ray–Hare test, we applied another hypothesis test - Aligned Rank Transform (ART) ANOVA [17]. The result are shown in Table 10 and Table 11. The magnitude of all p-value 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 10: 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 11: 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}

6.4 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 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 3 and Figure 5. The density plot for RQ1 shows large difference between native app and web app. For the Block1, there are some overlap areas between native app and web app while the plot still present difference between two groups. For the Block2, there is no overlap between two groups, therefore giving us the result representing large effect size.

7 DISCUSSION

As the statistical test results shown before, we can now answer our research questions. We first elaborate on two blocks in RQ1 separately. In Block 1, the p-value of Wilcoxon rank-sum test is less than 0.05, so we reject the null hypothesis that there is no energy consumption difference between the native app and their web version. For Block 2, we also accept the alternative hypothesis that there is difference between these two platforms. Compared to Block 1(big company), Block 2 (small company) has a smaller p-value, we can infer that big company pay more attention to balancing the performance of native apps and their web versions. However, the small company have less funds so they can only focus on developing their native apps. According to the box plot Figure 4 about platform, we also infer that the mean energy consumption of native app is less than their web version in both two blocks. The density plot for RQ1 shown in Figure 5 also reveals that web consumes more energy than native application.

For RQ2.1, we use Scheirer–Ray–Hare Test to investigate energy consumption difference between different platforms, and further take the network type into consideration. The results infer that the platform has main effects on energy consumption while the network type have no effect on energy consumption. The trend of density plot shown in Figure 7 is also an evidence that there is no significant difference between Wifi and 4g. Moreover, the test shows that platform and network type do have interaction effect on energy consumption.

For RQ2.2, we take two factors (router distance and platforms) into consideration, and Scheirer–Ray–Hare Test is also conducted to investigate the energy consumption difference between these two factors. For effect between different platforms, we derive the same results as the one in RQ2.1. However, the test result under router distance factor has a p-value bigger than 0.05 so we have a conclusion that it has no main effects on energy consumption. And these two factors do have interactions. The patterns between far and close shown in density plot are also similar.

The effect size gives us more evidence that the energy consumption between native apps and their web versions are different and small company block has a larger difference than big company block.

According to our results, for most companies, the energy consumption of their native app is less than their web version. In order to attract more users to download their applications, companies pay more attention to developing their native apps. However, there are still many people who use their web versions. It is wise to balance the energy consumption difference between native apps and their web version to increase the user experience of their web version. Developers should also take router distance and network types into consideration because it has interaction effect with platforms on energy consumption.

8 THREATS TO VALIDITY

We have analyzed the validity of our experiment based on four types of classifications, which were defined by Cook and Campbell [18]. We will describe these four types of classification of threats to validity of our experiment in the following sections:

8.1 Internal Validity

8.1.1 History. We collected our energy consumption data in 3 experiment processes. Each process contains two tests of 20 runs. And the time gap between each process were 2 days. In the first process we collected energy consumption data of 10 Android applications in two tests. Each test contains 20 runs of 5 Android applications developed by big or small companies. The time gap between the two tests are several hours. In the second process we collected energy consumption data of Android applications and web applications in situation that the device was in long distance to the router in two tests of 20 runs. In the third process we collected energy consumption data of Android applications and web applications in situation that the device was in 4G signal. Since the energy consumption of web applications is possible to change over days for the reason of webpage updates, history may be a threat to validity. We put the data collection of energy consumption of web applications from a certain research question in the same day. In this way we can reduce the effects of time for web applications within each research question.

8.1.2 Maturation. Maturation can be a threat to validity in our repetitions of tests. The current run might be effected 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 mintues' 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 effected by previous runs in our repeated trials.

8.1.3 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 same place.

8.2 External Validity

8.2.1 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's that meet the requirements described in section 4.1. Five of these applications have a download higher than 1 billion (large companies), and the remaining five have less than 1 billion downloads (small companies). The random selection process reduced the possibility to introduce biases. However, the filter process may affect the external validity and resulted 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.

8.2.2 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 difficult to reappear in 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 common-used browser in the real world. Therefore, our experiment was performed in a realistic environment.

8.3 Construct Validity

8.3.1 Definition of constructs. We used 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 GQM method, we mitigated the inadequate pre-operational explanation of constructs.

8.3.2 Mono-operation bias. Our experiment has only one independent variable which made it's possible to unable to represent the theory. However, for each research question, we use ten 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.

8.4 Conclusion Validity

8.4.1 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.

8.4.2 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.

9 CONCLUSIONS

We designed and conducted the experiment to investigate our main research question: if the energy consumption is different between applications and their web version with two other factors (network technologies and router distance) in this article. We can conclude that platform has the main effect on energy consumption according to our experiment results. Also, for both two blocks in RQ1, apps of the web version consume more energy than their native version. However, the other two factors (network types and router distance) for RQ2 have no significant effects on energy consumption, and only 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 to decrease energy consumption and help them 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 app and their web versions. If the dataset is large enough, it may allow us to conduct parametric tests which can produce more powerful results. It is also a good extension to regard different companies as factors so that we can determine if different companies' products have different energy consumption. At last, we can explore deeper to investigate if the number of network requests used by app and web version will influence the energy consumption.

REFERENCES

[1] A. Charland and B. Leroux, "Mobile application development: web vs. native," *Communications of the ACM*, vol. 54, no. 5, pp. 49–53, 2011.
[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] 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.
[5] 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
[6] 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.
[7] 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.
[8] 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
[9] C. Wohlin, P. Runeson, M. Höst, M. C. Ohlsson, B. Regnell, and A. Wesslén, *Experimentation in software engineering*. Springer Science & Business Media, 2012.
[10] C. Wohlin, P. Runeson, M. Höst, M. Ohlsson, B. Regnell, and A. Wesslén, *Experimentation in Software Engineering - An Introduction*. Kluwer Academic Publishers, 2012.
[11] 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)*, 2017, pp. 103–114.
[12] 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. [Online]. Available: https://github.com/S2-group/android-runner/blob/master/documentation/A_Mobile_2020.pdf
[13] 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.
[14] M. R. Harwell, "Choosing between parametric and nonparametric tests," *Journal of Counseling & Development*, vol. 67, no. 1, pp. 35–38, 1988.
[15] W. J. Conover, *Practical nonparametric statistics*. john wiley & sons, 1999, vol. 350.
[16] 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.
[17] 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.
[18] T. Cook and D. Campbell, "Quasi-experimentation: Design and analysis issues for field settings 1979 boston," *MA Houghton Mifflin*.