

Star-rating evaluation model for rating the energy-efficiency level of android google play apps

Abdullah Mahmoud Almasri¹, Luis Borges Gouveia²

¹College of Computer and Info Science, Prince Sultan University, Saudi Arabia

²Science and Technology Faculty, University Fernando Pessoa, Portugal

Article Info

Article history:

Received Aug 3, 2020

Revised Sep 6, 2020

Accepted Nov 4, 2020

Keywords:

Android apps

Energy-aware refactoring

Energy efficiency

Power-saving

ABSTRACT

The tremendous increase in smartphone usage is accompanied by an increase in the need for more energy. This preoperational relationship between modern technology and energy generates energy-greedy apps, and therefore power-hungry end users. With many apps falling under the same category in an app store, these apps usually share similar functionality. Because developers follow different design and development schools, each app has its energy-consumption habits. Since apps share similar features, an end-user with limited access to recharging resources would prefer an energy-friendly app rather than a popular energy-greedy app. However, app stores do not indicate the energy behavior of the apps they offer, which causes users to randomly choose apps without understanding their energy-consumption behavior. A review of the relevant literature was provided covering various energy-saving techniques. The results gave an initial impression about the popularity of the usage of two power-saving modes where the average usage of these modes did not exceed 31% among the total 443 Android users. To address this issue, we propose a star-rating evaluation model (SREM), an approach that generates a tentative energy rating label for each app. The model was tested on 7 open-source apps to act as a primary evaluation sample. To that end, SREM adapts current energy-aware refactoring tools to demonstrate the level of energy consumption of an app and presents it in a star-rating schema similar to the Ecolabels used on electrical home appliances. As per our results, SREM helped in saving 35% of smartphone energy.

This is an open access article under the [CC BY-SA](https://creativecommons.org/licenses/by-sa/4.0/) license.



Corresponding Author:

Abdullah Mahmoud Almasri

College of Computer and Information Sciences

Prince Sultan University

Riyadh, Saudi Arabia

Email: aalmasri@psu.edu.sa

1. INTRODUCTION

Users of Android smartphones have shown a growing demand for extending the smartphone's battery life [1-4]. One main concept that is shared between green computing and energy-saving research areas is to work toward avoiding the need for recharging resources instead of increasing the availability or efficiency of these resources, e.g., power banks, wireless charging, etc. Android smartphones currently are equipped with energy-saving solutions and features that are created by manufacturers and developers and then provided to end-users [5]. Recent studies have proposed several methodologies, models, frameworks, and approaches for saving energy. Efforts have been made to indicate hardware components that are considered as energy-greedy resources on mobile platforms (e.g., screens, network interfaces, sensors, etc.)

[6]. However, the effect of installed apps on a smartphone's battery is not easily understood by end-users [7]. The main contribution of our proposed solution is the design of a tool that can act as an effective decision support factor for end-users to have an initial indication of the energy-consumption behavior of an application before installing it. The core idea of the "before-installation" philosophy is simplified by the contradicting concept of installing the app and then having it monitored and optimized. Since processing requires power, avoiding the consumption of some power to conserve a larger amount of power should be our priority. So instead, we propose a preventive strategy that requires no processing on any layer of the smartphone. The nearest similar solution is the energy rating labels used on home appliances [8]. In the following, we show the proposed model as used in the Google Play® store to evaluate any Android app and then generate a tentative energy rating label for each Google Play app. We believe that our proposed approach of rating Android apps will enhance the role of end-users by giving them a much more clear, flexible, and effective way to customize the level of saving their smartphone batteries according to need. The approach will also inspire developers and app providers to come up with greedy multi-energy versions of the same apps to suit different categories of user needs.

2. BACKGROUND AND MOTIVATION

The main motivation of the research is the lack of user guidance for choosing suitable energy-efficient apps before installing and using the apps. The wide popularity of Android as a commonly used operating system among smartphones and smart devices, in addition to its major advantage of being an open-source platform, were the main factors motivating us to use Android apps as a primary area of research in our model. In the following, we propose an improvement to the energy-saving approaches currently used for Android apps especially during the pre-installation and usage stages. We formulated the research questions of this study as follows:

RQ1: How can we avoid sacrificing all or parts of a technology that are considered needed for specific smartphone users while still enabling users to extend the phone's battery life in a more efficient way (i.e., user-involved efficient preventive energy-saving)?

RQ2: How can we avoid wasting any amount of energy from a smartphone's battery while working on saving the energy of the battery (i.e., being preventive rather than a detective)?

3. RELATED WORK

Android smartphones come with many built-in features that are provided to end-users with a minimum level of participation [9]. These techniques and features follow certain philosophies. We were able to classify these into two approaches, detective and preventive, as shown in Figure 1. In the following, we provide an overview of the current approaches and their implementations and limitations.

3.1. Preventive vs detective battery-saving approaches on android smartphones

Solutions that are considered to follow the detective approach run on the system level of the Android smartphone and monitor the behavior of each app and component toward the battery [10]; these algorithms either act or warn the user or even kill an energy-greedy application. On the other hand, solutions that are provided by developers and manufacturers, which are considered under the preventive approach, are built-in battery-saving standards implemented on the hardware and kernel-level [11], e.g., energy star saver [12]. Another solution that is considered to follow the detective approach is battery optimizers, which are usually uploaded on Google Play as battery-saving applications. These applications do the same job that is achieved by built-in algorithms but with additional features since they run on the application and system level.

The second approach that we were able to classify is the preventive approach, which is also given by manufacturers and developers. One example of the preventive approach is the power-saving modes that the new Android smartphones are equipped with. These ready-made power-saving modes follow the preventive strategy. So instead of acting as a watch and monitor, they follow the idea of switching off most of the features and components that are considered as energy-greedy, e.g., GPS, flashlight, and Wi-Fi connection [13]. These components or features will be switched off to save the battery. Some of these power-saving modes may provide maximum discharging periods, which can reach weeks in some brands. At the same time, all this comes with the cost of using only basic and limited features of the phone. In other words, the screen brightness will be reduced or, in other extreme modes, it will be converted to grayscale, the Wi-Fi connection will be terminated, and the performance of the phone will be markedly reduced, keeping the option to exclude a small set of allowed apps. These power-saving modes that follow the preventive approach can be also customized by end-users to suit their own needs, but it involves many complications, which can

complicate the balance between saving energy and using the latest technology. Under the preventive energy-saving approach, we here demonstrate an example that highlights the key issue of having a limited capacity of decision-making among end-users when it comes to choosing from either high performance (full HD screen, continued GPS usage, etc.) and long battery life (minimizing the discharging time).

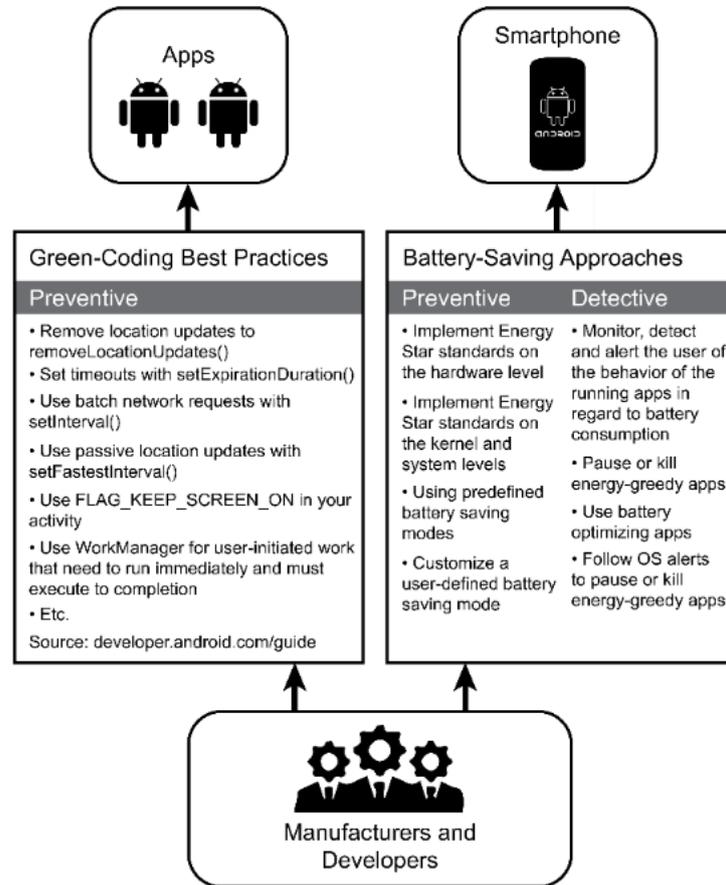


Figure 1. The high-level design of current battery-saving approaches implemented on android smartphones and applications

3.2. Android apps green coding practices

At the application layer, all solutions that are implemented at this level are considered to follow the preventive approach, since these applications are neither downloaded nor running on a smartphone, thus are considered to be in the “before install and use” zone [14]. These applications can run on a simulated environment using virtual Android platforms, but the actual management related to energy consumption is considered to be an estimation of their behavior [15]. The only stage that can handle solutions related to energy saving in the before install and use zone is during the development of the application. Speaking about coding and software, the standards that are followed are all considered best practices. These best practices are often used to implement quality standards related to performance, security, and, in our area, energy efficiency.

The official Android online resource lists several battery killers [16, 17]. These battery killers are either physical or virtual components in a smartphone, which usually consume a notable amount of energy whenever they are used. This online resource also proposes a set of best practices that are related to energy saving and power management during the development of an application. These practices propose minimizing the lines of code, avoiding the use of functions or loops that cause the application to run continuously, and putting restrictions on commands that take control of components considered to be battery killers [18]. As shown in Figure 1, the best practices used during development propose idling of the highest battery killers: the GPS, the screen, and the background processing. Following these recommendations and best practices are one way to ensure that an application is considered energy efficient. However, the effect of installed apps on a smartphone’s battery is not easily understood by end-users.

4. OVERVIEW BY EXAMPLE

A comprehensive example is a construction engineer who spends most of his time in open work areas that have either limited or no recharging resources and/or Wi-Fi connections. This engineer has an essential need to use a texting application similar to WhatsApp or Facebook messenger through a cellular data Internet connection, at the same time he needs to view a set of colorful architectural designs on his phone, and on top of all this, he needs to save as much as he can of his smartphone's battery life. The engineer has a set of options: The first is to enable a ready-to-use power-saving mode, which will affect all of the phone, as shown in Figure 2.

This effect may be in the form of switching off the cellular data connection to save the battery to place and receive ordinary GSM calls, which will deprive him of an Internet connection, and as a result not allow him to use a texting application. Another effect of enabling a power-saving mode is converting the screen into a grayscale mode, which also will not allow the construction engineer to view the colorful architectural designs. Another option is to customize the power-saving mode by enabling it and then excluding the cellular data connection, the colored screen, and all the other apps the engineer needs from the list of restrictions given by the power-saving mode. This will markedly minimize the efficiency of the power-saving mode in terms of minimizing the discharging time since it will be per app or feature usage, as shown in Figure 2.

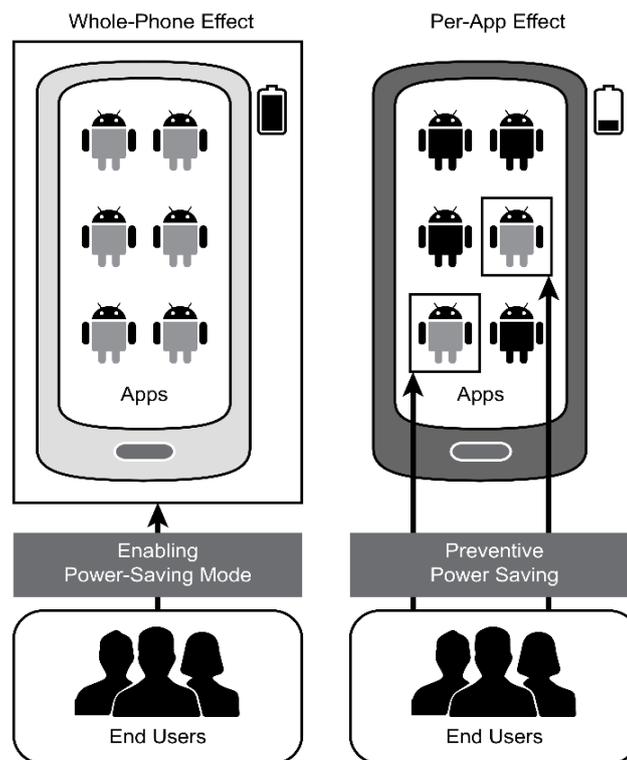


Figure 2. Whole-phone vs. per-app effects of preventive power-saving approaches given to users

5. PRELIMINARY STUDY RESEARCH METHODS

This paper proposes a novel way to enhance the current preventive energy-saving approaches for Android applications. To achieve this goal, the first step is to measure the efficiency level and popularity of the current preventive approaches. Knowing the level of user acceptance of the amount of flexibility and knowledge given to them while using the current preventive approaches is important for researchers and developers interested in improving these approaches. Specifically, if adding more flexibility or knowledge for end-users will significantly affect energy consumption, end-users should be provided with this additional level. A survey implemented among a sample of more than 443 Android smartphone users in 2019 got different results, which we first averaged and then classified by age groups from 16 to 60 years old. The survey had a primary question about using power-saving modes that Android smartphones are equipped with. Choices of answers were: Yes, uses the ready-made power-saving modes (YR); Yes, uses the customized

power-saving mode (YC); and No, never uses the power-saving modes (N). This part was expected to give a general view of the popularity of usage among these power-saving modes, therefore a good indication about their main contribution to extending battery life by end-users, as well as show their role in the big picture of green computing.

Other popular techniques offered by Android developers and manufacturers are battery optimizers and battery-saving applications. These applications run on the application layer and are intended to extend the battery discharge time. This technique is claimed to save energy under the detective approach, which follows the monitoring, control, and optimization of the behavior of each running application. To know the efficiency of power-saving applications, battery-saving tools, and battery optimizers, we implemented a short statistical and technical analysis on Google Play store applications. We picked a sample of the top 5 applications that offer the service of battery saving and management. We ordered the apps per the reviews, as they are the only indication of popularity and the number of downloads.

In the statistical part, we used PowerTutor as a power profiler to measure the amount of power consumption of each application per five minutes of continuous running. There are other ways to measure the amount of energy consumption, such as Msoon power monitor [19, 20], Trepn, and LEAP power measurement devices, but the main purpose was to give a general impression of the energy consumption by each app. The survey results were the most important source of inspiration to start looking for new approaches to save energy on smartphones since all current power-saving techniques follow either a preventive or detective power-saving approaches. The results of studying both examples under preventive and detective approaches are further demonstrated in the following section and then discussed concerning the background and the research questions.

6. RESULTS

The results of the preliminary study related to measuring the popularity of usage among power-saving modes were charted, as shown in Figure 3. From these results, we were first able to get an initial impression of the popularity of the usage of the two main categories of current power-saving modes. The average usage of these modes did not exceed 31% among the total 443 Android smartphone users. With this percentage taken into account and as far as the end-user has the option to either activate or avoid using a power-saving mode, this makes the concept of offering power-saving modes critically questionable in terms of real-life functionality and the main contribution to the big picture of extending battery life. However, since preventive approaches give the option either to enjoy the modern features of a smartphone or to enjoy a longer battery life, it always has to be offered as an optional solution. Therefore, we also studied the detective approaches, examples of which were battery optimizers and battery-saving apps offered in the Google Play store.

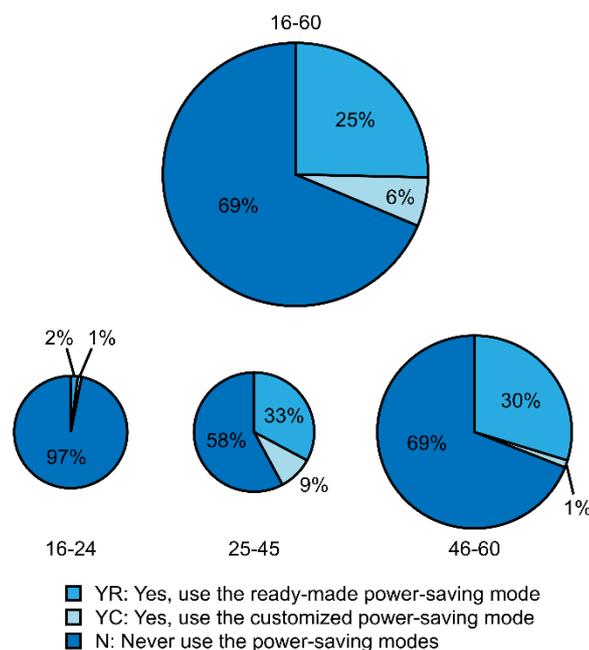


Figure 3. The popularity of the usage of power-saving modes among four end-user age groups

The results of the measurements generated by PowerTutor and Trepn are shown in Table 1. Additionally, by using the same power usage profilers to rank the power consumed by each running application, we were able to rank two power-optimizing applications, as both were ranked among the top three most power-consuming applications, as shown in Figure 4. Before going through the measurement results, these optimizers addressed a key issue related to the main concept of saving power. The main and critical issue is that they require power to run, which causes them to fail in delivering their main goal of saving energy. Put simply, monitoring apps on a running smartphone and announcing notifications will consume power from the smartphone's battery for the sake of saving the same smartphone's power. Plus, whatever runs on the application and/or the operating system (OS) layer of the phone consumes power from the same phone battery. Additionally, regarding these results, we were able to see that these applications consume a large amount of energy while running, and these results show a major limitation of this approach for energy-saving since there is a conflict of interest. This also makes this technique questionable in terms of its efficiency level compared to the main role of saving energy.

Table 1. Average power consumption per 5 minutes for 5 power-saving/optimizing apps

	Size	Rating	Reviews	Consumption Per 5 m
App1	10 MB	4.5*	8000000	~ 298.3 mW
App12	2.9 MB	4.4*	1000000	~ 165.7 mW
App6	15 MB	4.6*	189000	~ 785.9 mW
App11	7.6 MB	4.5*	135000	~ 454.6 mW
App2	12 MB	4.6*	118000	~ 652.1 mW



Figure 4. Using powertutor and trepn profilers to read the amount of energy consumed by two popular power-optimizing applications

7. THE MISSING PIECE OF THE PUZZLE

This research was aimed at finding the missing piece of the puzzle in the current framework of battery and energy saving in Android smartphones. Figure 5 demonstrates the current framework for the division of authorities and roles in energy-saving among Android smartphones. The most popular factors that smartphone users take into consideration when deciding which app to download and use are related to the main function of the app and its popularity. With the existence of other factors to consider, Android app stores allow users to choose from a variety of apps that share similar functions but give no indication as to the energy behavior of the apps. The area that shows our proposed solution is the area between applications that are not yet installed on the smartphone and the end-user, as shown in Figure 5. We propose that end users will act as decision-makers concerning the applications that are about to be downloaded on their smartphones after taking into consideration the level of power consumption for each app.

The main concept is to enhance the role of the end-users while selecting a reliable solution that follows the preventive approach, so end users will have a replacement option other than selecting a power-saving mode that will deactivate the modern features of the phone. This proposed enhancement will keep the same level of technology at the whole-phone level and will also allow end-users to select the best applications in terms of their energy-friendly features.

Referring to our previous overview example related to the field construction engineer, this will enable the engineer to select the level of functionalities by choosing a power-friendly app that still gives him more modern features while consuming less of his smartphone's battery compared to ordinary usage. So, the

simple scenario is that an end-user will be able to choose from the set of apps that share the same functionalities but have different energy consumption. In the following, we add the bridge between end-users and apps and then demonstrate the model that finally shows the strategy to be followed to rate the amount of energy consumption of an Android application.

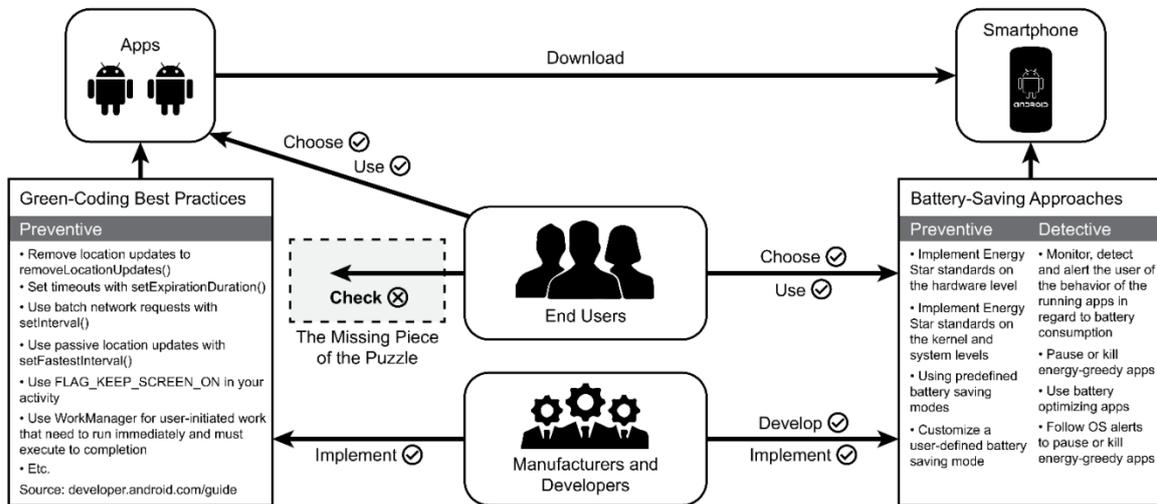


Figure 5. Current battery and energy-saving framework for Android smartphones

8. ENERGY STAR RATING SCHEMA: AN EFFICIENT DECISION SUPPORT TOOL FOR END USERS

The main contribution of our research is that we address the need to run a power-optimizing process on the OS and application layers of a smartphone. Since processing requires power, we focus on avoiding the use of energy stored in a battery to conserve that stored energy. Instead, we propose a preventive strategy that requires no processing in any layer of the smartphone and gives end-users the option to decide which apps to install after viewing their energy behavior. The nearest similar solution is the energy rating labels used on home appliances. Android users enjoy the way applications can be downloaded and used, but are also concerned about giving the green light to several flashing permissions. End users will also be able to see the star rating of each application available on the Google Play store, and then decide which applications to install and use.

8.1. Concept and challenges

The basic idea of the proposed solution is to benefit from the relevant proven results of researches that produced energy-friendly code restructuring or reformulating tools. The proposed solution merges these newly designed tools to use it as a set of baseline factors that will finally crystallize a concept of a reliable and trustful evaluation tool. This evaluation tool will be then used as an x-ray belt to evaluate any Android application in terms of its level of energy efficiency. The tool is supposed to be a hidden built-in feature added to Google Play Store which will show the level of energy efficiency of an app to all users of Google Play Store. Similar to the current user satisfaction start-rating schema which is shown beside each app. We preferred to use the current refactoring tools as baselines to our tool because of the availability of the source code on Google Play which makes it an extremely valuable resource for these refactoring tools to be implemented on most popular apps. So, the idea of bringing the source code of an app that is available on Google together with the latest refactoring tools is the primary contribution of this research.

However, the tool which we are proposing can be flexible to allow future baseline factors to be included in the evaluation process. The proposed solution in its simplified and final form is similar to the Echo-Star Schema or the “energy efficiency information sticker” which is placed on electrical home appliances. We will try to demonstrate the level of energy efficiency of each app in a similar star rating schema. This rating will be shown beside each app on Google Play aside from the ordinary reviews of current users. With this, we intend to enhance the authority of an Android app used to be able to decide which application to use based the level of energy efficiency which is calculated using verified and tested tools.

The additional new part of our contribution here is that the Android users will be using a new preventive strategy which will not cost any waste of energy. The users will be standing at the door of their

smartphones and allowing in only energy-efficient apps rather than letting whatever apps in and then trying to find a way to balance the energy consumption which will also cost energy consumption. For the technical part, the proposed tool is based on the principle of simple mathematical comparison. So, we first measure the maximum amount of energy consumption for the application by running it with its fullest capacity for a specified period on a simulated Android environment. Then we pass the source code of the app among the available refactoring tools to generate an energy-friendly version of the code. After that, we measure again the maximum amount of energy consumption for the application by running it with its fullest capacity for the same period in the same simulated Android environment. Finally, the greater the difference between the two readings, the weaker the energy efficiency index for the application is, or in other words, fewer stars.

Conversely, the smaller the difference between the two readings, the more the original code of the application shows adherence to the energy-friendly development practices, which means more stars. The top question which comes in mind is why not to easily provide the users with an energy-efficient version of the app which was generated after passing it through the refactoring tools instead of demonstrating its default energy efficiency level? The answer to this question is based on the code handling policy which is followed by Google Play in addition to the agreements between the developers and Google which avoids implementing alterations to the source code of an app except for scanning it for security and privacy breaching loopholes, so usually the code is uploaded and provided as is to end-users after in addition to notifying them about the required permission for the app to function correctly. This also raises a very big concern in terms of the need for the application.

Some applications that are currently running cannot be replaced by others because of their popularity, e.g., WhatsApp, Twitter, Snapchat, etc. Even if the star rating showed these applications as energy greedy, users will still install and use them on their smartphones. To address this concern, we believe that a new area of inspiration is now open for developers to provide different versions of applications: basic, light, or full. Each version will have a different level of energy vs. modern features, and end-users can choose both based on their own needs. Also, this will encourage developers to apply whatever recommendations are available on their apps to provide the highest energy-friendly ratings. In addition to the above, we proposed this option to end-users to be able to choose the best energy-friendly app from the list of apps that shares the same high-level goal, for example, texting apps, music players, voip apps etc. Therefore, going through the usefulness of each app is beyond the scope of our research. To be more clear, considering the echo-star rating sticker which is placed on home appliances, even-though appliances form the same category do not perform similar tasks the sticker does not go through the usefulness of each home appliance e.g. a washing machine of model X performs additional features (tasks) than a washing machine from model Y. But finally, the echo-star rating sticker shows only how much energy-friendly are both machines without going through the usefulness of each machine.

8.2. Approach

The main concept of our approach is to bring together developers, app repositories (Google Play), and researchers who have proposed automated energy-aware approaches to restructuring Android apps and those who have developed energy profilers. All of these parties will act as inputs to generate a knowledge-based schema that will help end-users to decide which applications to choose and install. An approach that previous research has proven to be efficient in saving energy through an automated framework is code refactoring [21]. The previous research followed either an anti-pattern or pattern-based refactoring approach [22]. Since we need a comparable reliable factor and for the model to be interoperable, we used one tool from each approach. The tool, called energy-aware refactoring approach for mobile apps (EARMO), proposed by Morales *et al.* [23], follows a novel anti-pattern correction approach that accounts for energy consumption when refactoring mobile anti-patterns. The results generated by this tool were used as a primary factor for our study. Another tool that uses the refactoring approach is Leafactor, proposed by Cruz *et al.* [24], which refactors the source code to follow a set of patterns known to be energy efficient. Here we propose a flexible multicriteria star-rating evaluation model (SREM) to generate tentative energy rating labels for Google Play store apps by adding the refactoring approach as the first criterion to be used for the rating process. The model, shown in Figure 6, shows where the tools are to be used in the process.

At the first stage, the source code of an Android application is provided. The source code is then installed on an Android platform before measuring its power consumption. The energy measurements are generated by profilers, which is the preferred solution since they allow finer-grained measurements and also because the experiments can be reproduced, which is not always possible with hardware solutions; PowerTutor was adapted to automate the profiling procedure. The amount of power consumption that is first reported by PowerTutor is recorded as E1 mW/time. After that, the app goes through the refactoring process, which is expected to refine the code to make it more energy-efficient. Next, the app is run again with its new refactored code to measure its new power consumption, E2 mW/time. Finally, a simple comparison can then

show the difference and be scaled to an equivalent 1–5 stars. Compared to E1, as E2 shrinks after refactoring, fewer stars are given to the app. Whenever E2 stays the same or changes slightly compared to E1, more stars are given to the app. Based on the previous results, we can now help end-users to have an initial indication of the energy-efficiency of the app they are about to install and use. We believe that the SREM approach will enhance the role of end-users to act not only for energy-saving techniques but to also participate as decision-makers by choosing what apps suit them in terms of energy consumption.

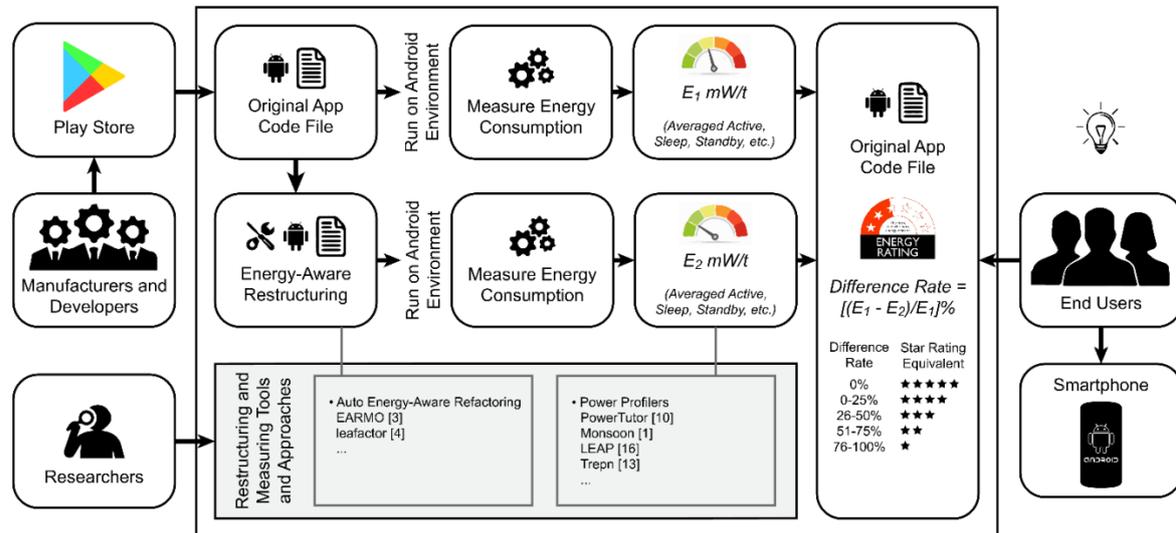


Figure 6. Star-rating evaluation model (SREM)

As Figure 6 shows, we also kept the model open and flexible for any additional tools or approaches that can improve the power consumption measurements or the approaches used for converting the source code of any app to be energy-efficient, and both improvements will increase the resolution of the comparison and improve the rating process. Although there is a large body of work on the energy consumption of Android apps, and research on saving energy suggests estimating the energy usage of an app, compared to our approach, most of these techniques do not compare apps to their lighter version in terms of power consumption. Our approach leverages the rating process to obtain the energy consumption of Android apps more efficiently.

9. EVALUATION

Since the SREM follows a comparison-based approach to rate apps, in addition to our model tending to combine and leverage the use of previously evaluated approaches, what comes next is to evaluate the outcomes of the whole model by knowing the total amount of energy saved after implementing the SREM. We conducted an initiatory evaluation of the SREM to demonstrate its ability to rank apps according to differences in power measurements before and after using the reconstructing approaches. To generate ground-truth estimates, we picked 7 open-source apps to act as a primary evaluation sample. Since we are proposing inter-usage of current automated energy-aware app restructuring and measuring approaches, we first needed to re-emphasize the efficiency of the current approaches. We used one approach as an automated energy-aware refactoring approach.

Refactoring was proven to reduce the amount of energy consumption according to Morales *et al.* [23], who followed a novel anti-pattern correction approach that accounts for energy consumption when refactoring mobile anti-patterns, EARMO. Another approach is Leafactor, proposed by Cruz *et al.* [24], which refactors the source code to follow a set of patterns known to be energy efficient. Table 2 demonstrates the results of using EARMO and Leafactor with the set of applications that we prepared for the evaluation. The first column refers to the app names coded by abbreviations. The second column refers to the amount of consumption before the refactoring per 30 minutes of continuous exhaustive usage (abbreviated by ECBR). The third column, which has two sub-columns, refers to the amount of power consumption of the app after implementing the two refactoring approaches per 30 minutes of continuous exhaustive usage (abbreviated by ECAR). The two sub-columns show the use of EARMO and Leafactor. The fourth column shows the energy

consumption of the app after averaging the consumption measurements taken from the two approaches. Then we show the amount of change and the change rate. The rate is calculated according to the equation represented in the SREM. So whenever the rate increase is large, it indicates that the refactoring approach had to wipe a good number of patterns related to energy-greedy processing. On the other hand, whenever the percentage indicates a small change, it shows that the refactoring tools did not have to go through a lot of effort to remove unwanted patterns.

Table 2. Results of using energy-aware refactoring approach for mobile apps (EARMO) and leafactor as inputs to SREM with 7 apps to generate star ratings

App	ECBR	ECAR		Average ECAF	Change Rate	Star Rating
		Leafactor	EARMO			
App1	895 mW	787 mW	795 mW	795 mW	13%	*****
App2	450 mW	150 mW	198 mW	198 mW	101%	*
App3	321 mW	150 mW	120 mW	120 mW	138%	*
App4	977 mW	850 mW	750 mW	750 mW	22%	****
App5	820 mW	550 mW	498 mW	498 mW	56%	***
App6	604 mW	540 mW	498 mW	498 mW	16%	*****
App7	650 mW	600 mW	620 mW	620 mW	7%	*****

To evaluate the efficiency of the proposed model, an alternative is to be chosen for each none-energy-efficient application. Table 3 shows an example of an Android smartphone with 10 applications installed. The same table proposes energy-efficient alternatives for each installed app. These energy-efficient alternative apps were chosen based on average similar features or tasks and listed under the same app category on Google Play Store. Energy consumption measurements were generated using PowerTutor. Finally, the percentage of saved energy is also presented which clearly shows a total of 35% saving if the user of this specific smartphone decides to go for the proposed energy-efficient apps. Since users are currently kept without guidance in this aspect, guiding users towards energy-efficient apps through energy rating is the main contribution of our proposed model.

Table 3. Results of using energy-aware refactoring approach for mobile apps (EARMO) and leafactor as inputs to SREM with 20 apps to generate star ratings

Installed App	App Average	Alternative Energy-Efficient App	Alternative App	Energy Saved %
	Energy Consumption/min		Average Energy Consumption/min	
RainViewer	~ 20 mW	Rain Alarm	~ 8 mW	60%
Weather Live	~ 20 mW	World Weather	~ 15 mW	25%
Free VPN & Security	~ 18 mW	VPN Free-Betternet	~ 14 mW	22%
PicArt Photo Editor	~ 18 mW	Snapseed	~ 15 mW	17%
SkyScanner	~ 18 mW	Tripadvisor Hotel, Flight	~ 14 mW	22%
Otter	~ 18 mW	Transcripser	~ 15 mW	17%
Hotspot Shield	~ 17 mW	VPN Free-Betternet	~ 9 mW	47%
Udemy	~ 17 mW	edX: Online Courses	~ 9 mW	47%
Be Closer	~ 17 mW	GeoZilla	~ 10 mW	41%
Jollychic	~ 17 mW	Rain Alarm	~ 8 mW	53%
Total Consumption of Installed Apps	~180 mW	Total Consumption of Alternative Apps	~117 mW	35%

10. DISCUSSION

The study made some key assumptions about user behavior. For example, it was generally assumed that users will use the most power-efficient option, which is the fundamental aspect of any future calculations performed for measuring the energy reduction. However, the study does not consider the impact of brand loyalty on app choice and app usage. Several studies examine the determinants of brand loyalty in mobile applications. For example, Kumar *et al.* [25] highlight that holistic visual aesthetic dimensions can influence mobile app loyalty, demonstrating these findings through applying Kaplan's information processing model from environmental psychology and integrating it into the technology acceptance model. Considering Kumar *et al.*'s findings, it can be presumed that switching applications might not be a chosen approach if the proposed new apps differ significantly in terms of visual aesthetics dimensions.

Besides, app satisfaction, the user's intention to continue using the app, and the hedonic benefits obtained from using apps (such as consisting of app aesthetics and enjoyment) are the direct antecedents of

intention to recommend the app to another user [26]. This affirms the findings presented above regarding the importance of aesthetics and enjoyment of use and illustrates that in terms of app use, battery performance has no observed impact on the user's intention to seize using an application or recommend it to a friend. App usability, as demonstrated by Baek and Yoo's [27] study is a multidimensional construct, consisting of 13 items in five factors: user-friendliness, personalization, speed, fun, and omnipresence, while extended brand use can also be a result of brand attributes of the developer corporation, such as its warmth and competence [28].

Therefore, even if the user recognizes the value in making the recommended app swaps, they could continue using the battery-hungry applications, citing other in-app factors and features, which have ultimately led to them becoming loyal customers, or continue using the apps due to factors, related to the company, which has developed it. Overall, this demonstrates that while the proposed approach is logically-sound based on the efficiency of the battery, other determinants influence app choice and user behavior, which can obstruct the translation of the efficiency savings, cited in the findings to a practical setting. As also previously highlighted, different users have vastly contrasting energy management behaviors, which relates to how battery-saving is approached and also how a battery is discharged [29].

The perceived importance of app loyalty and app choice versus battery performance improvement is also not well understood, with further research on this topic needed. While the present research findings suggest that the individuals from the group with the highest attention towards battery performance can benefit from switching to other applications, which can perform the same functions at a lesser expense to the phone's battery, it is unknown to what degree they would be willing to make the swap. In other words, while this coping strategy is considered by the authors least invasive to the lives of the individuals amongst the group, it is understood that other coping strategies exist, which might be preferred by different users as they consider the least invasive and most convenient. The recommended approach remains objectively the most efficient approach based on environmental protection, battery performance optimization, and personalization of the approach to the user's individual app use.

It is also recognized that energy savings do not always translate to extended battery life—an insight, affirmed within Kim *et al.*'s study [30]. Specifically, the authors write: 'energy savings do not always translate to longer smartphone battery life and that evaluating any savings plan must be based on battery consumption, not energy used'. In some cases, the manufacturers of batteries could have produced them with a certain life in mind. In all cases though, the current research has demonstrated a reduction of battery consumption, which leads to less frequent charges required, thus generating better behavior concerning green computing and energy use. Whether this translates to an extension of the lifetime of the device or the battery remains a point of debate in academic science.

Finally, it is recognized that there might be in some cases applications that cannot be replaced, even though there are more efficient, optimized applications doing the same service, with the issues going beyond the functional characteristics of the app. Referring back to the discussion illustrated as part of the literature review, regarding the recent pandemic, many users have now installed applications that support their day-to-day communications with their co-workers, the change of which is in most cases arguably impossible, as it is a decision, taken at a higher corporate level. This example serves to demonstrate that while users have ownership of their devices, due to the intertwined nature of smartphone devices, applications, and services in our daily lives, there is sometimes an inability to manage the applications on the phones with complete ownership of the decision-making. Having certain applications, even if they are battery-damaging could be a means to sustain our roles as employees or individuals as part of a digital society. Examples of advanced applications that are considered unexchangeable to certain users are those offering automatic road distress visual inspection and human hand motion recognition [31, 32].

11. IMPLICATIONS OF FINDINGS

The implications of this research can be considered as relative importance for policy, practice, theory, and further research. In terms of the policy, the current research highlighted the rising tension between the field of technology and environmentalism, specifically how the rise in technology consumption negatively impacts the environment. The proposed approach arguably presents an opportunity for two groups of stakeholders—users and developers of energy-efficient apps. For users, this study demonstrates that aside from a personal psychological affirmation, there is no reason to continue using an application that is severely battery-depleting continuously, especially if leading a primarily-outdoor-based, busy lifestyle. The few of the hindrances of doing so are a faster battery life depletion, environmental impact, higher electricity bills, caused by unnecessary charging of the device, ultimately leading to a shorter lifespan of the device. The study affirms the benefits of switching to applications that do not automatically operate in the background or have an 'always-on' location tracking function. From an app developer standpoint, the results from this study

can be used as a promotion technique, given that the apps developed fall into the category of applications, deemed as battery efficient. Also, the proposed solution serves the developers as it shows the energy rating without revealing any technical details to the app(s) users which is also the proposed privacy policy of the model.

12. SIGNIFICANCE OF THE STUDY

This study benefits users and app developers, as it systematically demonstrates the importance of engaging in more efficient battery saving behaviors. From a user standpoint, this can be through using any of the listed energy-efficient apps on an Android apps store. For developers, the study highlights the negative impact of energy-hungry apps, which could lead to a negative brand association of the user, who could relate the app to reduced battery performance, following the app's installation. Considering the illustrated findings concerning the rising importance of smartphones in the lives of people, as well as the lagging advancements made in the technical development of higher battery capacity of smartphone devices, it can be speculated that the battery concerns of users will continue intensifying in the future, with more and more people paying attention to their battery performance. This trend, if observed, paired with the proposed approach to battery optimization, would give rise to app comparison platforms and services, which will favor developers and applications, which create apps that are battery-sparing.

13. CONCLUSIONS AND FUTURE WORK

The study is considered successful in achieving the research objectives at the problem identified and remaining diligent in the created procedural model. These results demonstrate high potential for the development of a complete application that can be deployed as assistance to users in navigating the app landscape through making choices that are responsible and beneficial for both themselves and the environment. Importantly, this study also highlighted the limitations of the current model, namely the lack of consideration of how brand attributes can influence app choice, potentially obstructing users to benefit from more battery-sparing applications and services on their devices. The illustrated limitations highlighted a variety of avenues for future research, which can examine the psychological or other barriers users have in engaging in battery-saving behavior.

The study is considered significant for both users and app developers, with it being informative for the former group and potentially opportunistic for the latter, provided that they engage in behaviors that align with the principles of green computing and energy conservation. Although end users are given the ability to choose from a limited number of power-saving approaches, app stores give no indications about the energy behavior of the apps they offer. This energy-related mystery causes end users to randomly choose apps without understanding their energy-consumption behavior. We introduced the SREM to tackle this issue, an approach that estimates the energy-friendly level of Android apps and demonstrates it in a star-rating schema similar to the energy-efficiency labels placed on home appliances. Our aim in this work is to generate the number of stars for each application from the difference between the amount of ordinary power consumption and the amount of power consumption after applying an energy-aware restructuring approach to the app. We relied on the refactoring approach as a previously evaluated efficient approach to restructuring apps to energy-aware versions.

Android app stores can use the SREM to enhance the role of end-users in deciding which apps meet their energy consumption needs. The SREM will also inspire developers and app providers to come up with multiple energy-greedy versions of the same app to suit the needs of different categories of users and rate their apps. To evaluate the efficiency of the model as an addition to Google Play store apps, which will act as labeling to guide end-users, a future study is to be implemented to know the effect of the SREM on the decisions of end-users and, as a result, on the main goal of extending the battery life of smartphones. The survey will address the rates of those who are still selecting power-greedy apps and those who decided to switch to power-friendly apps.

ACKNOWLEDGEMENTS

The authors graciously thank the College of Computer and Information Sciences, Prince Sultan University, and the Faculty of Science and Technology, University Fernando Pessoa for their scientific and administrative support.

REFERENCES

- [1] H. Anwar, et al., "Evaluating the impact of code smell refactoring on the energy consumption of Android applications," in *2019 45th Euromicro Conference on Software Engineering and Advanced Applications (SEAA)*, Kallithea-Chalkidiki, Greece, 2019, pp. 82-86.
- [2] A. Almasri and A. Sameh, "Saving energy on smartphones by categorizing users," in *2019 2nd IEEE Middle East North Africa Communication Conference (MENACOMM)*, Manama, Bahrain, 2019, pp. 1-4.
- [3] A. Sameh and A. Al-Masri, "Smartphone preventive customized power-saving modes," *International Journal of UbiComp*, vol. 8, no. 1, pp. 1-15, 2017.
- [4] L. Zhang, et al., "Accurate online power estimation and automatic battery behavior based power model generation for smartphones," *2010 IEEE/ACM/IFIP International Conference on Hardware/Software Codesign and System Synthesis (CODES+ISSS)*, Scottsdale, AZ, 2010, pp. 105-114.
- [5] M. Couto, et al., "GreenDroid: A tool for analyzing power consumption in the android ecosystem," in *2015 IEEE 13th International Scientific Conference Informatica*, Poprad, Slovakia, 2015, pp. 73-78.
- [6] R. W. Ahmad, et al., "A survey on energy estimation and power modeling schemes for smartphone applications," *International Journal of Communication Systems*, vol. 30, no. 11, p. e3234, 2017.
- [7] T. A. Dao, et al., "TIDE: A user-centric tool for identifying energy-hungry applications on smartphones," *IEEE/ACM Transaction on Networking*, vol. 25, no. 3, pp. 1459-1474, 2017.
- [8] "Labelling Energy Rating," *Energyrating.gov.au*, 2020. [Online]. Available: <https://www.energyrating.gov.au/about-e3-program/labelling>.
- [9] A. Almasri, "Preventive customized strategies for saving energy on smartphones," M.Sc. Thesis, Prince Sultan Univ., Riyadh, Saudi Arabia, 2015.
- [10] L. Li, et al., "SceneMan: Bridging mobile apps with system energy manager via scenario notification," in *2017 IEEE/ACM International Symposium on Low Power Electronics and Design (ISLPED)*, Taipei, Taiwan, 2017, pp. 1-6.
- [11] X. Chen and Z. Zong, "Android app energy efficiency: The impact of language, runtime, compiler, and implementation," in *2016 IEEE International Conference on Big Data Cloud Computing (BDCloud), Social Computing and Networking (SocialCom), Sustainable Computing and Communications (SustainCom) (BDCloud-SocialCom-SustainCom)*, 2016, pp. 485-492.
- [12] *EnergyStar.gov*, "ENERGY STAR," 2020. [Online]. Available: <https://www.energystar.gov>.
- [13] A. Almasri and A. Sameh, "Rating Google-Play Apps' energy consumption on Android smartphones," in *2019 2nd IEEE Middle East and North Africa Communication Conference (MENACOMM)*, Manama, Bahrain, 2019, pp. 1-6.
- [14] J. A. Montenegro, et al., "What do software developers need to know to build secure energy-efficient Android applications?" *IEEE Access*, vol. 6, pp. 1428-1450, 2018.
- [15] A. Banerjee, et al., "EnergyPatch: Repairing resource leaks to improve energy-efficiency of Android apps," *IEEE Transaction on Software Engineering*, vol. 44, no. 5, pp. 470-490, 2018.
- [16] *MakeUseOf*, "Android Battery Killers: 10 Worst Apps That Drain Phone Battery," 2020. [Online]. Available: <https://www.makeuseof.com/tag/android-battery-killers-drain-worst-apps/>.
- [17] *Android Developers*, "Android Developers," 2020. [Online]. Available: <https://developer.android.com/>.
- [18] L. Donatiello and G. Marfia, "Modeling the energy consumption of upload patterns on smartphones and IoT devices," *IEEE Communication Letter*, vol. 22, no. 11, pp. 2258-2261, 2018.
- [19] *Monsoon Solutions Inc*, 2020. [Online]. Available: <https://www.monsoon.com/LabEquipment/PowerMonitor/>.
- [20] D. Singh and W. J. Kaiser, "The atom leap platform for energy-efficient embedded computing," Center for embedded network sensing, University of California, USA, 2010.
- [21] A. Banerjee and A. Roychoudhury, "Automated re-factoring of Android apps to enhance energy-efficiency," in *2016 IEEE/ACM International Conference on Mobile Software Engineering and Systems (MOBILESoft)*, Austin, TX, USA, 2016, pp. 139-150.
- [22] X. Li and J. P. Gallagher, "A source-level energy optimization framework for mobile applications," in *2016 IEEE 16th International Workshop Conference on Source Code Analysis and Manipulation (SCAM)*, Raleigh, NC, 2016, pp. 31-40.
- [23] R. Morales, et al., "EARMO: An energy-aware refactoring approach for mobile apps," in *40th IEEE/ACM International Conference on Software Engineering (ICSE 2018)*, Gothenburg, Sweden, 2018, p. 59.
- [24] L. Cruz, et al., "Leafactor: Improving the energy efficiency of Android apps via automatic refactoring," in *2017 IEEE/ACM 4th International Conference on Mobile Software Engineering and Systems (MOBILESoft)*, Buenos Aires, Argentina, 2017, pp. 205-206.
- [25] D. Kumar, et al., "Influences of 'apps cape' on mobile app adoption and m-loyalty," *Journal of Retailing and Consumer Services*, vol. 45, pp. 132-141, 2018.
- [26] C. Xu, et al., "A customer value, satisfaction, and loyalty perspective of mobile application recommendations," *Decision Support Systems*, vol. 79, pp. 171-183, 2015.
- [27] T. Baek and C. Yoo, "Branded App Usability: Conceptualization, Measurement, and Prediction of Consumer Loyalty," *Journal of Advertising*, vol. 47, no. 1, pp. 70-82, 2018.
- [28] Y. Fang, "An app a day keeps a customer connected: Explicating loyalty to brands and branded applications through the lens of affordance and service-dominant logic," *Information & Management*, vol. 56, no. 3, pp. 377-391, 2019.
- [29] S. Hosio, et al., "Monetary Assessment of Battery Life on Smartphones," *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*, 2016, pp. 1-12.

- [30] M. Kim, et al., "Measuring Variance between Smartphone Energy Consumption and Battery Life," *Computer*, vol. 47, no. 7, pp. 59-65, 2014.
- [31] T. Siriborvornratanakul, "An Automatic Road Distress Visual Inspection System Using an Onboard In-Car Camera," *Advances in Multimedia*, vol. 2018, pp. 1-10, 2018.
- [32] C. Kerdvibulvech, "Human Hand Motion Recognition Using an Extended Particle Filter," *Articulated Motion and Deformable Objects*, pp. 71-80, 2014.

BIOGRAPHIES OF AUTHORS



Abdullah Mahmoud Almasri holds a bachelor of science degree in Information Systems and a Master of Science degree in Software Engineering from Prince Sultan University, Saudi Arabia. Senior Lecturer and researcher of Computer Science and Information Systems at Prince Sultan University since 2015. His current research interests include Mobile Computing, Power-Aware Applications, and Systems Design.



Luis Borges Gouveia holds a Habilitation in Engineering and Industrial Management from the University of Aveiro, Portugal, and a Ph.D. in Computer Science from the University of Lancaster, UK. Full Professor at University Fernando Pessoa, Porto, Portugal where is the coordinator of the Ph.D. program in Information Science. His main interests are related to how computers and digital applications can impact people, and include novel application to provide better sustainable technology usage.