Ph.D. Dissertation in Engineering

Essays on User-Preference-Based Optimization Methods for Overcoming the Resource Limitations of Smartphones

스마트폰의 자원 제약을 극복하기 위한 사용자 선호 기반 최적화 방법론 연구

August 2016

Graduate School of Seoul National University

Technology Management, Economics and Policy Program

Baseem Taher Othman Al-athwari

Essays on User-Preference-Based Optimization Methods for Overcoming the Resource Limitations of Smartphones

지도교수 Jörn Altmann

이 논문을 공학박사 학위논문으로 제출함

2016년 08월

서울대학교 대학원
협동과정 기술경영경제정책전공
Baseem Taher Othman Al-athwari

바심 오트만 알아트오리의 박사학위 논문을 인준함
2016년 08월

위 원 장_____ (인)

부위원장_____(인)

위　원_____(인)

위　원_____(인)

위　원_____(인)

Abstract

Essays on User-Preference-Based Optimization Methods for Overcoming the Resource Limitations of Smartphones

Baseem Taher Othman Al-athwari
Technology Management, Economics, and Policy program
College of Engineering
Seoul National University

There is no doubt that the realized technological development in smartphones technology and its applications during the last few years had resulted in a remarkable increase in the smartphone usage, not only for simple services such as calling and sending messages, but also for performing a huge number of activities such as social networking, web browsing, emailing, video watching, and gaming. However, running such activities implies a heavy workload on smartphones and results on significant resources consumption (memory, processing, communication, and energy).

Despite its attractive design (i.e. small size, and light weight), diverse range of capabilities, and multi-functionality, smartphone still have limited resources, such as battery energy, processing power, network bandwidth, and

storage capacity. These constraints of mobile resources result in limited use of smartphones and boost the need for solutions to overcome these limitations.

Battery lifetime of smartphone is one of the most critical limitations of smartphones. It has always been a concern not only for smartphone users but also for the manufacturers. Despite the significant efforts to improve battery technology, the advance in the battery technology has not been able to keep pace with the rapidly growing demand for power consumption for mobile activities.

Because smartphone battery lifetime depends on usage behavior, it is necessary to understand how users use up the battery of the smartphone. Many researchers have made great efforts on usage behavior of smartphone users. All of them found that users have their own usage pattern. This diversifies among users revealed the need of understanding the user behavior. Without understanding user behavior, it is not possible to clear understand the impact of any optimization on user experience or the mobile power consumption.

Recently, Mobile cloud computing (MCC) has been introduced to overcome the resource restrictions of smartphones by enabling them to offload the computational intensive applications on resourceful clouds instead of running them locally on the device. This is referred to as computation offloading. However, offloading the computation to the cloud definitely introduces costs for the mobile user, based on the exact resources consumed (energy; bandwidth; money; etc.). Therefore, offloading should only occur if the offloading cost is smaller than the cost of local. In addition to the cost

issues, there are some user context issues such as the remaining energy in his smartphone, and his location, which does not make it always desirable to offload the application execution to the cloud.

The overall objective of this study is to find solutions for overcoming the limitations of resource-constraint smartphones considering two perspectives: energy consumption, and smartphone applications execution. To address these resources restrictions of smartphone, this dissertation focuses on two key issues. First: the energy consumption optimization was studied by taking into account the user preferences to maximize the users' utility from the energy remaining in his smartphone battery. In particular, a utility-based energy consumption optimization model was proposed which considers the user preferences with respect to energy allocation to the two types of smartphone application uses (i.e., on-device application use and cloud-based application use). The working of the model is demonstrated by conducting both quantitative and qualitative research techniques including collecting usage data from real smartphone users, and in-person interview.

Second issue deals with multi-criteria optimization for supporting computation offloading of smartphone applications to the mobile cloud computing (MCC). Here, an offloading decision making model that minimizes the offloading cost subject to multiple constraints including application, smartphone, network, and cloud constraints while considering the user context and the user preferences is proposed. The different costs (i.e., time cost, energy cost, communication cost, and computation cost), which are incurred by executing the application locally on the mobile device or by

offloading the application to the cloud), are integrated through the use of user preferences in making this multi-criteria offloading decision. Although it can be assumed that users can consider all various combinations of those factors and make a good decision, the frequency of those decisions will be cumbersome for the user. In addition, there is no-specific linear equation which can describe the relationship between all those factors that influence the offloading decision. In this regard, the neural networks are used to model the non-linear interaction among these multiple factors. A Deep Neural Network (DNN) was trained using offloading decisions examples made by the user based on the proposed offloading decision model to support the user decision in making optimal offloading decision in the long run. The potential of neural networks for making offloading decision is evaluated through a use case.

The main contribution of this research rises from the new approaches that this work presents in dealing with the overcoming the resource restrictions of smartphone. Most of the previous studies about smartphone usage and energy consumption as well as the computation offloading are frequently focusing on developing technical solutions and rarely consider the user as a part of their solutions. Such solutions unable to capture user intention and preferences and hence fail to match the user expectations from using the smartphone. Therefore, this work opens the doors for new research area in smartphone technology that will contribute to further improvements in the smartphone platforms.

Keywords: Smartphones, Apps, Energy consumption optimization, Utility function, Usage behavior, Cloud computing, Application classification, Energy allocation, Computation Offloading, Mobile Cloud Computing, Cost Model, User Preferences, Offloading Decision Making, Neural Networks, Optimization.

Student Number: 2012-31306

Contents

List of Tables

List of Figures

Chapter1. Introduction

1.1 Background

With the rapid growth of the smartphones market and its applications, smartphones increasingly became an essential part of human life. In our daily lives, we use mobiles to do many activities such as social networking, web browsing, emailing, video watching, and gaming. Because of the unique characteristics of smartphones, they have become powerful ultra-portable personal computers supporting not only communication but also running a variety of complex, interactive applications (Banovic, 2014).

Considering the battery constraint of the smartphone, user often cares a lot that his smartphone battery to last for long time to maximize his utility from the amount of the energy saved on his smartphone. Moreover, it might became important to make the smartphone last as long as possible. This, for example, can be achieved by reducing the execution of some applications. Then, the user can execute the most important applications on his smartphone.

There are many other situations, in which users can make energy preserving decisions. These decisions of a smartphone user can range from choosing which applications to use, how to use an application, to how long the application should be used. For example, it could consider when to make a phone call, what to browse on the Internet, and which video to watch (Tarkoma, Siekkinen, Lagerspetz, & Xiao, 2014). Consequently, the amount of energy consumed per application is different and the overall energy consumption depends on the user behavior.

In the recent years, the concept of offloading the computation and data to the cloud has been introduced to address the limitation of mobile devices by using the resources provided by the cloud rather than the mobile device itself to run the mobile application (Church, 2015). There is a significant amount of computation offloading frameworks which were designed with different objectives. Among the popular objectives are: minimizing energy consumption, reduction in execution time, and reducing network latency.

1.2 Problem Description

Maximizing their utility is strong intensive for smartphone users to save the energy. Users' interest towards saving energy corresponds to their convenience. Therefore, it is within the users' interest to optimize the energy consumption of his mobile devices to prevent the battery running out of energy on a critical moment (Heikkinen, Nurminen, Smura, & HäMmäInen, 2012).

As the battery lifetime of smartphone depends on usage behavior, it is important to capture statistics on application usage and energy consumption to understand and model user preferences. In addition, because users may change their usage behavior, such as when the remaining energy in the battery got low, it is also important to understand the motivation behind their behavior (Froehlich et al., 2007).

Many researchers have made great efforts on usage behavior of smartphone users. All of them found that users have their own usage pattern. Significance difference in user's' usage behaviors make a general purpose energy optimization solution unsuitable for mobile devices (Pasricha et al., 2015). This diversity among users revealed the need of customizing the energy consumption optimization to the

users. In order to customize the energy consumption to the user it is necessary to understand the user behavior and define a model that can describe the user behavior. Despite the quantitative difference in the usage behavior among users, the model can use the qualitative similarities to describe the behavior of all users (Froehlich et al., 2007).

Better understanding of user behavior helps in multiple ways in developing new mechanisms that better match user expectations (Falaki, Mahajan, et al., 2010; Heikkinen et al., 2012). Technical solutions should learn and adapt user behavior to effectively improve user experience or energy consumption (Falaki, Mahajan, et al., 2010). Without understanding user behavior, it is not possible to clearly understand the impact of any optimization on user experience or the mobile power consumption (Shye, Scholbrock, & Memik, 2009).

Recently, Mobile cloud computing (MCC) has been introduced to overcome the limitations of smartphones by enabling mobile devices to offload the computational intensive applications on resourceful clouds instead of running them locally on the device. This is referred to as computation offloading. However, transferring the execution of a mobile application to the cloud requires not only communication between the mobile device and the cloud but definitely introduces costs for the mobile user, based on the exact resources consumed. It might consume additional energy, bandwidth, and money to cover the cost for communication and cloud computations. Therefore, in some cases, the offloading cost could exceed the cost of local execution as a result of the additional resource utilization in the offloading process. In addition to the cost issues, there are some user context issues

such as the remaining energy in his smartphone, and his location, which does not make it always desirable to offload the application execution to the cloud.

1.3 Main Objectives and Research Questions

The overall research objective of this study is to find solutions for overcoming the limitations of resource-constraint smartphones considering energy consumption, and smartphone applications execution. Accordingly, this study consists of two main essays, each of which is written as separate research paper.

The objective of the first essay (Chapter 3) is on the energy consumption optimization of smartphones taking into account the user preferences to maximize the users' utility. A utility-based energy consumption optimization model is the focus which considers the user preferences with respect to energy allocation to the two types of smartphone application uses (i.e., on-device application use and cloud-based application use). In particular, the objectives which are investigated are:

(1) Design a user utility-based energy consumption optimization model for on-device application uses and cloud –based application uses.

(2) Understanding of how the user utility can be maximized, subject to the remaining energy in the battery and the user preferences for different types of application uses.

In this regard, the study aims at addressing the following questions:

- How the user preferences for different types of applications uses can be modeled taking into consideration energy consumption aspect?

- How the user's utility can be maximized, subject to the remaining energy in the battery and the user preferences for different types of application uses?
- How can the model be applied in the real world?

The Objective of the second essay (Chapter 4) focuses developing an offloading decision strategy for the computation offloading which can help in making optimal offloading decision based on different costs of the offloading while considering user context and preference. However, no-specific linear equation can describe the relationship between all those factors that influence the offloading decision. Therefore, the more specific objective is to develop an offloading decision making strategy for computation offloading which combines all the different cost factors as well as considers the user context and preferences. The objectives which are investigated are:

(1) Development of an comprehensive offloading decision strategy for the computation offloading which can help in making optimal offloading decision based on different costs of the offloading while considering user context and preference.

(2) Examination a method for supporting users in making offloading decisions.

In this regard, the study aims at addressing the following questions:

- How can we combine time cost, energy consumption cost, communication cost, and computation cost, and user context for the offloading decision?
- How can the offloading decision be supported through an optimization method that is based on the user preferences?

- What is the performance accuracy of such an optimization method?

1.4 Data and Methodology

The user is the main source of the data used in this study. For the first essay, because smartphone's battery lifetime is dependent on usage behavior, it is necessary to model the user preference for the different types of application uses with respect to their energy consumption. Based on that, the first essay focuses on the energy consumption optimization of smartphones taking into account the user preferences to maximize the users' utility. In particular, a utility-based energy consumption optimization model is proposed which considers the user preferences with respect to energy allocation to the two types of smartphone application uses (i.e., on-device application use and cloud-based application use). The shape of the user utility function is defined by the Cobb-Douglas function. The working of the model is demonstrated by using real smartphone usage data for the simulation which has been collected from a study with a real smartphone users to explore their preferences for using the smartphone applications through the past usage pattern data regarding energy consumed by the two types of application uses, and through their perceptions for the energy allocation to the two types of smartphone application uses. For setting the parameters of the proposed utility-based energy consumption mode, we choose Android OS based smartphones as it is popular, holding the biggest market share in the smartphone industry. According to data from the International Data Corporation (IDC) Worldwide Quarterly Mobile Phone Tracker, Android dominated the market with an 82.8% share in 2015 Q2. We recruited 10 participants all of them are living in Korea. In general, Internet

penetration rate in Korea is very high and WiFi networks are deployed widely. We asked the participants to install *eStar Energy Saver* application on their own smartphones .The application is easy to access and it provides information about the percentage of energy consumed by the applications launched by the user as well as system applications. It also shows the time duration for each application usage. As our study focuses on the usage behavior/preference, we only focus on the applications launched by the user.

The participants were asked to send their battery monitoring data by the *eStar Energy Saver* application for 10 complete battery usage periods. Battery usage period refers to one complete battery usage without connecting the mobile to the charger (one complete discharging period). It begins when the smartphone battery is full, and ends when the smartphone battery is discharged.

Based on the data of past behaviors that have been obtained by monitoring the users' application uses, we calculated the energy consumed through on-device application uses and cloud-based application uses for each usage period for each user. Using the obtained values for the past consumption of energy for 10 battery usage periods, we calculated the average amount of battery energy consumed through on-device application uses and cloud-based application uses for each user. Therefore, based on obtained values, the long-term preference for the energy consumption for each user is modeled using the utility function.

To maximize the utility from the energy remaining in the smartphone battery, we conducted an interview with the participants and asked them to express their perceived costs for allocating energy to on-device application uses and to cloud-based application uses.

Participants indicated that their perceived cost is varied according to the energy remaining in their smartphone battery. They noted that if the level of the remained energy in the battery goes low they tend to give up some amount of energy consumption by one type of application uses for the other preferred type of application uses. This analysis led us to emphasize the importance of adopting the rate of substitution (RS) concept as a measure to explore the utility maximization combination of energy allocation to on-device application uses and to cloud-based application uses.

In the second essay the focus is on the development of an offloading decision strategy for the computation offloading. This strategy could help in making right decisions as to whether to perform computation offloading or not based on different costs of the offloading while considering user context and preference. The different costs (i.e. time cost, energy cost, communication cost, and computation cost), which are incurred by executing the application locally on the mobile device or by offloading the application to the cloud), are integrated through the use of user preferences in making this multi-criteria offloading decision. Therefore, the model consists of four components: constraints, cost estimation model, user context, and decision making. The cost estimation component takes into account the constraints and calculates the different costs including time cost, energy cost, and monetary cost for both local and offloading executions and passes it to the decision making component.

Although it can be assumed that users can consider all various combinations of those factors and make a good decision, the frequency of those decisions will be cumbersome for the user. In addition, since there is no-specific linear equation

which can describe the relationship between all those factors that influence the offloading decision, the neural networks is used to model the non-linear interaction among these multiple factors. In this regard, a Deep Neural Network (DNN) was trained using offloading decisions examples made by the user based on the proposed offloading decision model to support the user decision in making optimal offloading decision in the long run. The potential of neural networks for making offloading decision is evaluated through a use case. Using data about offloading decisions examples made by user (labels) based on those different cost factors and user context, two neural networks with different architectures were trained and evaluated. We refer to the user offloading decision as our target variable, because it is what we need to examine. The neural networks with those 8 input values representing the cost and context factors were trained and tested using a frequency of 2000 offloading decisions made by the user (labels). We consider the offloading decision as a binary problem (0 represents not offload, and 1 represents offloading). The NN is trained using 80% of the training data labeled by the user while 10% data is used for validation. The generalization capability of the network is evaluated with the remaining 10% data.

1.5 Contribution

This work provides some additional interesting understanding of smartphone usage behavior, and it will contribute to the line of studies assessing the effect of user preferences on energy consumption optimization of smartphones as well as the overall usage experience. It can be used by smartphone users to obtain the maximum

utility from the remaining energy in their smartphones battery. And also it can be used by the smartphone manufactures and developers to enhance the smartphone's platform and applications design. In detail, we make the following contributions: (1) a utility-based energy consumption optimization model; (2) the demonstration of the application of the optimization model to two types of application uses (on-device application uses, cloud-based application uses) with a real case study; (3) an offloading decision making algorithm is proposed, which aims at minimizing different aspects of the offloading cost while considering user context (e.g., his mobility and the energy remained in his smartphone) and preferences;. (4) Neural networks that support the user in making the complex optimal offloading decisions based on different cost and context factors.

1.6 Thesis Outline

This study consists of two main essays, each of each is written as a separate research paper, yet both are integrated in one here. The thesis was organized into five chapters as it can be seen in Figure 1.Chapter 1 gives an overall introduction to facilitate this integration. It presents the background, the main objectives and research questions; data and methodologies used in this study; and ends with the study contribution. Chapter 2 presents the start-of-the-art related to the smartphone usage, human-battery interaction; as well as the different computation offloading frameworks which developed in the literature to address different objectives. It also presents description of the neural networks. Chapter 3 is about utility-based smartphone energy consumption optimization for cloud-based and on-device

application uses. In this chapter, In particular, a utility-based energy consumption optimization model is proposed which considers the user preferences with respect to energy allocation to the two types of smartphone application uses (i.e., on-device application use and cloud-based application use). The working of the model is demonstrated by conducting a real case study with real smartphone users to explore their preferences for using the smartphone applications through the past usage pattern data regarding energy consumed by the two types of application uses, and through their perceptions for the energy allocation to the two types of smartphone application uses. Chapter 4 presents study on user-preference-based optimization for making multi-criteria offloading decisions in mobile clouds. In this Chapter an offloading decision making model that minimizes the offloading cost subject to multiple constraints is proposed. The different costs and context factors are integrated through the use of user preferences in making this multi-criteria offloading decision. The potential of optimal offloading decision by the user is supported through the using of neural networks. Chapter 5 consists of the discussion and conclusion; main findings of the study, and discusses the implications of this dissertation. Study limitations, and suggestions for future researches are also included in this chapter.

Chapter 1 Introduction

- Background
- Problem Description, Main Objectives and Research Questions
- Data and Methodology, Contribution
- Thesis outline

Chapter 2 State-of-the-Art

- Mobile usage
- Human-Battery Interaction
- Mobile Cloud Computing
- Computation offloading
- Neural Networks

Chapter 3: Utility-Based Smartphone Energy Consumption Optimization for cloud-Based and On-Device Application Uses

Objectives:-Design a utility-based energy consumption model for on-device application uses and cloud-based application uses.

-understanding user utility maximization.

Research Questions: **i)** How the user preference for the different types of application uses can be modeled taking into consideration energy consumption aspect? **ii)** How the user utility can be maximized subjected to the battery remaining energy and user preferences for different types of application uses? iii) How the model can be applied in the real world?

Methodologies: i) Theory of consumer behavior (Utility function); **ii)** Case study: Quantitative and Qualitative

Contributions: i) Utility-based energy consumption optimization model. ii) Model Application to real case

Chapter 4: User-Preference-Based Optimization for Making Multi-Criteria Offloading Decisions in Mobile Clouds

Objectives:- developing a comprehensive offloading decision strategy for the computation offloading based on different costs and context factors.

-Examining ta method for supporting users in making offloading decisions

Research Questions: **i)** How can we combine time cost, energy cost, monetary cost and user context for the computation of offloading decision? **ii)** How can the offloading decision be supported through an optimization method that is based on user preferences? **iii)** What is the performance of such optimization method?

Methodologies: -i) Cost Modeling; **ii)** Neural Networks.

Contributions: i) User-preferences-based offloading decision algorithm. **ii)** Using the Neural Network to support the user for making complex optimal offloading decision.

Chapter 5 Conclusion & Implications

i) By taking into account the user preferences, the energy consumption of smartphone can be adjusted to maximize the user utility. ii) Neural Network can be used to support the user in making optimal offloading decision based on complex combinations of costs and context factors.

Implications :Smartphone developers and manufactures should recognize that: i) there is a need to implement a learning mechanism to monitor user behavior and learn his preference and provide the user with a feedback that could help him to adjust his usage behavior and take proactive measures to allocate the energy consumption between the different types of the applications uses ii) should give the users control over specific tasks such as whether to execute the application locally on device or offload it to the cloud in order to improve the usage experience for the smartphone user..

Figure 1. Thesis Structure

Chapter 2. State-of-the-Art

2.1 Mobile Device Usage

In the recent years, we have seen a steady increase in the human dependency on smartphone devices in his daily life. Mobile devices have emerged rapidly from simple communication device into multifunctional information and communication device(Ferreira, Goncalves, Kostakos, Barkhuus, & Dey, 2014). With the increase in their functionality and diversity of use, smartphone devices are predicted to be the dominant future computing devices with highest user expectations for running computationally intensive applications like those of powerful desktop, notebook, or PCs machines (Shiraz, Gani, Khokhar, & Buyya, 2013). A key to the smartphone and other portable devices such as PDAs and tablets is the mobility feature which makes it possible to use them without having to connect them to the charger all the time(Ferreira, Ferreira, Goncalves, Kostakos, & Dey, 2013). During the past years, the internet-based applications and service were only available for device with limited mobility, such as laptops, but recently became accessible through those smartphone devices (Heikkinen et al., 2012). Another key feature of the smartphone devices is the user interaction patterns and interfaces. Unlike other devices such a PC or desktop in which the user typically interacts with application using a pointer device or keyboard, applications on smartphone devices most often receive user input through a touch screen or keypad events (Pasricha, Donohoo, & Ohlsen, 2015). However, in spite of all the developments in the recent years, the mobile applications on the latest generation of smartphones and tablets are still restricted by

the resource limitations of smartphones such as battery lifetime, CPU, memory and storage capacity (Shiraz et al., 2013).

Over the past decade, there is a significant number of real-world user studies of mobile device use. To understand usage behavior, researchers have adopted a range of methodologies. One such method is interview users for qualitative data regarding their mobile device usage ((Banovic, Brant, Mankoff, & Dey, 2014). Another method is the use of logger application by installing a monitoring application on user's smartphone. Some researchers have focused on developing tools for gathering mobile usage data. Such tools have provided the researchers with the capabilities to study actual smartphone usage for large scale in the number of users and over long period (Church, Ferreira, Banovic, & Lyons, 2015; Falaki, Mahajan, et al., 2010; Heikkinen et al., 2012; Kang, Seo, & Hong, 2011; Pasricha et al., 2015; Shye et al., 2009). In some other works, researchers combine usage monitoring with questionnaires ((Froehlich, Chen, Consolvo, Harrison, & Landay, 2007; Ahmad Rahmati, Qian & Zhong, 2007; Ahmad Rahmati & Zhong, 2010) .

Although the user interview method in studying usage behavior doesn't scale well for large number of users, it has the benefit that it can provides answers about all kind of issues that are not readily measurable (Heikkinen et al., 2012). On the other, the main advantage of the monitoring method through a logger application is that, it can be used to conduct large-scale studies across users form different places and collect large amount of data for long period. However, it cannot be used to capture the user intention (Church et al., 2015; Froehlich et al., 2007; Heikkinen et al., 2012). In smaller scale mobile studies, researchers tend to combine questionnaires with the logging method (Church et al., 2015). Combining the two

methods can help to check whether the user's actions matched their stated behavior or not (Heikkinen et al., 2012).

Despite, the significant amount of research about mobile usage with different sizes of samples , and from different populations, with each study focused on different aspect of mobile usage, most researchers agree that studying mobile device usage is still very challenging(Church et al., 2015) . Another common finding is that all users have unique usage patterns. Even one study reveals a change in user type (Tarkoma et al., 2014). Therefore, such studies contribute to broader understanding of smartphone usage behaviors as well as providing insights about future areas of mobile related researches (Church et al., 2015).

2.2 Mobile Cloud Computing

2.2.1 Definition

With the explosive growth of mobile applications, the mobile cloud computing (MCC) has been emerged as a new computing paradigm for mobile applications. It provides mobile users with data processing and storage services in clouds(Dinh, Lee, Niyato, & Wang, 2011). The usage of MCC can leverage the limitations of mobile devices (storage and processing power) by allowing the execution of computation intensive applications on low resources mobile devices (Fernando, Loke, & Rahayu, 2013) .

There is no single widely accepted view of MCC and there are several existing definitions of mobile cloud computing. In common, the term MCC refers to the execution of the mobile application on the cloud server. It can be defined as an

infrastructure where both the data processing and data occur outside of the mobile device(Dinh et al., 2011). MCC can also be defined as a combination of cloud computing and mobile web (Christensen, 2009; Liu, Moulic, & Shea, 2010) , which is the most popular tool that provides the mobile user with access to application and services on the internet. MCC is also described by (Chetan, Kumar, Dinesh, Mathew, & Abhimanyu) as a concept that aims at using cloud computing techniques for processing and storage data of mobile devices. Alternatively, (Soyata, Ba, Heinzelman, Kwon, & Shi, 2013) defines the MCC as optimization of an objective function which involves the execution of mobile application within the cloud platform . Examples of objective function are minimizing the energy consumption and/or the application response time.

2.2.2 Generalized Mobile Cloud Computing Architecture

2.2.2.1 Remote Cloud

General architecture of MCC is composed of smartphone, the internet and computational cloud. Smartphone which runs applications is connected to the internet through cellular network or Wi-Fi to deliver the mobile user's requests to the cloud. In the cloud, requests are processed and sent back to the smartphone users with the corresponding cloud services. In this architecture, the mobile device acts like a thin client connected to the remote cloud server(Fernando et al., 2013). Although cloud-based architecture provides a diverse range of resources and services that are available on demand in the cloud with high computation power(Abolfazli, Sanaei, Ahmed, Gani, & Buyya, 2014), however, the limitation of this kind of mobile cloud computing architecture is that, cloud servers are not

always available (Ghasemi-Falavarjani, Nematbakhsh, & Ghahfarokhi, 2015). In high mobility conditions, disconnections to the cloud server can happen quite often. Also when the network connection is poor, offloading the computation to the remote cloud introduces latency(Hung, Shih, Shieh, Lee, & Huang, 2012). Moreover, as the offloading the computation to the cloud involves costs in terms of energy, time, communication; the computation offloading cost can exceed the local execution cost. Therefore to is not always feasible to offload the computation to the remote cloud server.

2.2.2.2 Cloudlet

The cloudlet concept proposed by (Satyanarayanan, Bahl, Caceres, & Davies, 2009) as another approach to mobile cloud computing. In this architecture the mobile application is offloaded to a nearby server known as cloudlet. The cloudlets are comprised of several resource-rich computers deployed inside the local network. The main objective of the cloudlet is to reduce latency and minimizes the data transport cost by offloading the computation to nearby available computers using WLAN(Ahmed, Gani, Khan, Buyya, & Khan, 2015; Pal, 2015) .These cloudlets-based MCC doesn't require internet connection. In spite of these advantages provided by the cloudlet, the cloudlet limits the mobility of users (Ghasemi-Falavarjani et al., 2015). Also, Considering their local network , cloudlet providers need to prevent outside users form accessing the resources in their cloudlets(Ahmed, Gani, Khan, et al., 2015). Moreover, the deployment of the cloudlets requires incentives for cloudlet service providers, well-defined charging policy as per usage

basis, and more resource to provide the on-demand services and resources to the MCC users (Ahmed, Gani, Khan, et al., 2015) .

2.2.2.3 Mobile-ad-hoc

Another approach is to offload the computation among neighboring mobile devices making up a mobile peer-to-peer network as in(Huerta-Canepa & Lee, 2010; Koukoumidis, Lymberopoulos, Strauss, Liu, & Burger, 2011). This approach doesn't require any local server for the deployment of the cloudlet where both clients and servers are mobile devices. In this architecture, all mobile devices shared their idle resources among each other. Connectivity opportunities in this type of mobile cloud are fully depends upon the users' common activities .It assumes that all mobile users are cooperative. However, this is not applicable in the real case as users may not have common goal which motivates them to share their resources with others. In addition, user's mobility is considered in this kind of communication. However, such mobility is considered challenging due to the sporadic contact of the users and their unpredictable behaviors (Karamshuk, Boldrini, Conti, & Passarella, 2011; Pal, 2015). Due to such mobility, it is possible that mobile devices might not be present within the ad-hoc mobile network during offloading process (Shikfa, 2010). Moreover, as the service providers in this kind of mobile cloud are the mobile devices which have limited battery energy capacity, it is possible that a mobile device runs out of energy and dropped from the network. (Ghasemi-Falavarjani et al., 2015).

2.2.2.4 Hybrid MCC

To overcome the limitations of three types of the aforementioned mobile clouds, some of the existing frameworks combine the capabilities of remote cloud servers and local cloudlet servers in a hybrid platform (Cardellini et al., 2013; Kovachev, Yu, & Klamma, 2012; Magurawalage, Yang, Hu, & Zhang, 2014; Marinelli, 2009; Rahimi, Venkatasubramanian, Mehrotra, & Vasilakos, 2013; A. Ravi & Peddoju, 2015; Wei, Fan, Lu, & Ding, 2013; B. Zhou, Dastjerdi, Calheiros, Srirama, & Buyya, 2015). Accessing the resources and services in the remote cloud introduce arises a number of issues, such as latency resulted from the WAN delay, jitter, congestion, and even connection failures (Satyanarayanan et al., 2009). On the other hand, as the cloudlets have limited resources, they cannot scale well with the increasing number of the users within the cloudlet network (Ahmed, Gani, Khan, et al., 2015). The deployment of hybrid platform enables the mobile devices to take benefit from the nearby cloudlet servers, and cloud servers. As a result, it helps to provide the flexible execution for the mobile applications such that runtime critical components of the application can be executed on the nearby low-latency cloudlet and delay-tolerant components of an application can be migrated to be run on the remote high-latency cloud servers.

2.3 Computation offloading

2.3.1 Definition

The concept of offloading the computation and data to the cloud is used to address the limitation of mobile devices by using the resources provided by the cloud rather

than the mobile device itself to run the mobile application (Fernando et al., 2013). Computation offloading is a solution to augment the mobile's capabilities by migrating computation from mobile phones to more powerful and resourceful computing servers located in the cloud (Kumar, Liu, Lu, & Bhargava, 2013). The process of moving computation from mobile device to the cloud is called computation offloading. It is typically used to boost the computational capability of a resource constrained mobile device. In mobile Cloud computing (MCC) environment, the computation offloading is defined as the mechanism of migrating resource-intensive computation from mobile device to the resource-rich cloud (Enzai, Idawati, & Tang, 2014).

2.3.2 State-of-the-Art Computation Offloading Frameworks

Recently, several computational offloading frameworks have been proposed for offloading computational intensive mobile applications partially or entirely to the cloud. The latest developments in the computation offloading frameworks in MCC have aimed at augmenting the resources of mobile devices by leveraging the resources and services of the cloud (Ahmed, Gani, Sookhak, Ab Hamid, & Xia, 2015). Most of the existing offloading frameworks focus on what components of the application to offload, how to offload the components and where to offload the intensive component of the application (Shiraz, Sookhak, Gani, & Shah, 2015).

The state of the art application offloading frameworks are designated to address different objectives. Among popular objectives are minimizing energy consumption, minimizing execution time, minimizing the monetary cost, and reducing the network

latency. Table 1 presents a comparison between the state-of-the- art computation offloading frameworks based on the offloading objectives.

Table 1. Comparison of Computation offloading frameworks based on their Objectives

Offloading Objective	**Literature**							
	ThinkAir, (Kosta et al., 2012)	MAUI (Cuervo et al., 2010)	Wolski at al., 2008	Cuckoo, (Kemp et al., 2012)	CloneCloud, (Chun et al., 2011)	Phone2Cloud (Xia, F. et al ., 2014	Calling the Cloud (Giurgiu et al., 2009)	Zhang et al., 2010
Minimizing Energy consumption	O	O		O	O	O		O
Minimizing Execution time(Improve the performance)	O	O	O	O	O	O	O	O
Minimizing monetary cost								O
Reducing Network Latency							O	

ThinkAir (Kosta, Aucinas, Hui, Mortier, & Zhang, 2012) propose ThinkAir, , a dynamic and adaptive framework that exploits the parallelism to simultaneously execute multiple offloading methods using multiple virtual machines (VM) images to reduce the application execution time and improve the power of the mobile cloud. The ThinkAir framework logs the energy consumption, execution time, and network conditions, which are used to decide whether a method should be offloaded or not. However, due to the intensive profiling mechanism, the offloading decision of the ThinkAir framework is complex.

MAUI: (Cuervo et al., 2010) propose an energy-aware mobile application offloading framework . MAUI supports fine grained offloading to enhance energy saving with negligible load on the programmer. It performs cost-benefit analysis by profiling each method in an application. Specifically; MAUI continuously collects essential data such as energy consumption, CPU utilization, and network bandwidth condition, at runtime. The collected data by the profiler is then used to decide whether the method should be executed locally or remotely. Although MAUI significantly improves the battery life of a mobile device and incorporates the user mobility, it does not address the transmission latency and scalability, and does not provide the QoS features(Ahmed, Gani, Khan, et al., 2015). In addition, the MAUI mechanism for of run time application profiling and solving involves additional computing resources.

(Wolski, Gurun, Krintz, & Nurmi, 2008) consider the bandwidth aspect and present a framework for making computation offloading decisions to improve the performance. They have examined various decision strategies for offloading in a grid computing case. The authors predict the time required to execute the computation locally and remotely and use offloading to minimize total execution time. They do not consider the energy aspect of offloading as they used the bandwidth between the local and remote endpoints as the bottleneck.

Cuckoo (Kemp, Palmer, Kielmann, & Bal, 2010) propose a partial offloading framework for mobile applications to the cloud or nearby cloudlet servers. The Cuckoo aims at minimize the energy consumption and execution time of the application. Although the decision parameters of the Cuckoo are decided at run time, it takes static decisions for offloading that are context unaware.

CloneCloud (Chun, Ihm, Maniatis, Naik, & Patti, 2011) propose an elastic execution framework that enables mobile devices to offload part of the application computation to the cloud. The framework employs dynamic profiling to collect data used in making the offloading decision, and static analysis to partition the application. The main goal of portioning is to optimize the overall execution cost. The execution cost comprises energy cost and execution time. The energy consumption cost consists of CPU activity, display state, and the network state. The computation cost takes values from the clone cost variables when the method runs on the clone of the mobile device, or from the mobile device cost variables. The migration cost sums the individual migration cost of those invocations whose method have migration points.

phone2Cloud (Xia et al., 2014) have developed a computation offloading-based system for energy saving on smartphones called phone2clou. The main aim of this system is to reduce the energy consumption and enhance the performance through reducing the execution time. They implemented the prototype of the phone2cloud on Android and Hadoop environment.

Calling the cloud (Giurgiu, Riva, Juric, Krivulev, & Alonso, 2009) developed a middleware framework that supports dynamic partitioning of mobile applications to be offloaded to the cloud. The objective of this framework is to minimize the interaction latency between the mobile device and the cloud server while considering the exchange data overhead. The Calling cloud framework reduces the memory consumption, communication cost, and interaction time. However, due to the dynamic analysis, profiling, synthesis, runtime partitioning and offloading, it employs compute-intensive offloading process. The framework also requires

continuous synchronization which keeps mobile device in active state for the whole session of distributed platform.

(X. Zhang, Kunjithapatham, Jeong, & Gibbs, 2011) propose an elastic application model that provides a middle framework for mobile applications. Their model based on the dynamic partitioning of the application into multiple components called weblets which are replicated across multiple clouds. The offloading decision is based on complex cost model which incorporates four attributes including energy consumption, performance attributes, monetary cost, and security and privacy. Thereby, making the offloading decision process compute-intensive. The framework implements a resource-intensive mechanism for runtime application partitioning and for the migration of weblets between the smartphone and remote cloud nodes. Thereby, it requires additional computation resource utilization on smartphone during application profiling, dynamic runtime partitioning, weblet migration and reintegration, and continuous synchronization with the cloud server node for the entire duration of the application processing.

2.4 Neural Networks

It is well known that neural networks (NNs) are effective tools for modeling the non-linear interactions among multiple variables (Haykin & Network, 2004; Principe, Euliano, & Lefebvre, 1999). Deep Neural Networks (DNN) can learn complex non-linear functions from training data (the concept of learning from data). “ A Neural Network (NN) suitably trained with a limited number of configurations can be successfully used in the optimization procedure, and it can evaluate the

objectives values instead of the costly assessment of the numerical procedure" (Carcangiu, Fanni, & Montisci, 2011).

Neural Networks are so widely studies by different research communities in different contexts, such as Computer Science, Electrical Engineering, Biology, and Psychology). Recently, deep learning using neural networks have been used to achieve state-of-the-art performances in a wide variety of tasks. These include (but not limited to) speech recognition(Chan, Jaitly, Le, & Vinyals, 2016; Dahl, Yu, Deng, & Acero, 2012; Deng & Togneri, 2015; Hinton et al., 2012; Mohamed, Dahl, & Hinton, 2009; Seide, Li, & Yu, 2011; P. Zhou, Jiang, Dai, Hu, & Liu, 2015), image classification (Cichy, Khosla, Pantazis, Torralba, & Oliva, 2016; Ciresan, Meier, Gambardella, & Schmidhuber, 2010; Cireşan, Meier, Masci, Gambardella, & Schmidhuber, 2011; Donahue et al., 2013; Jarrett, Kavukcuoglu, Ranzato, & LeCun, 2009; Krizhevsky, Sutskever, & Hinton, 2012; Shah, Bennamoun, & Boussaid, 2016; Simonyan & Zisserman, 2014), natural language processing, and bioinformatics.

A neural network is composed of number of nodes, or neurons, connected by links. Each link has a numeric weight associated with it. Weights are modified during the learning. Each neuron has a set of input links from other neurons, a set of output links to other neurons, a current activation level, and means of computing the activation level at the next step in time, given its inputs and weights (Russell, S., et al., 1995). Figure 2 shows a typical neuron.

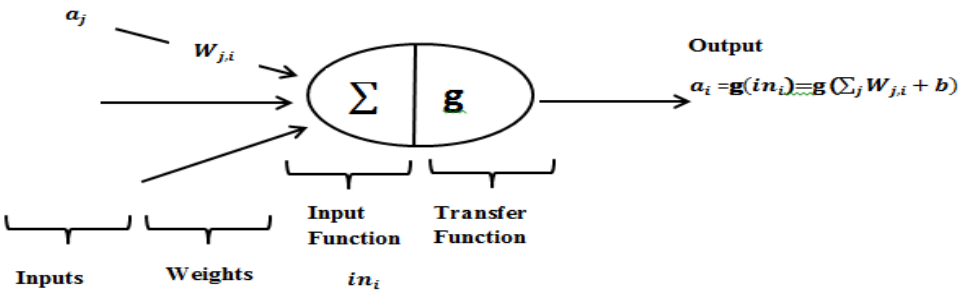


Figure 2. A neuron

Notation:

$\boldsymbol{a_j}$: Activation value (output) of neuron j (also the input of the neuron *i)*

$\boldsymbol{W_{j,i}}$ **:** weight on the link from neuron j to neuron *i*

g: Transfer function (also called Activation function)

b:Bias

$\boldsymbol{in_i}$**:** weighted sum of inputs to neuron *i*. All inputs to a neuron are combined into a single number using the following weighted sum:

$$in_i = \sum_j w_{j,i}\, a_j + b \qquad (2\text{-}1)$$

The weights and bias terms are modified during network training. The activation function transforms this weighted sum into output value. The idea is that, each neuron calculates his output from its net input based on inputs from its neighbors, but without the need for any global control over the set of neurons as a whole (Russell, S., et al., 1995)

Chapter3. Utility-Based Smartphone Energy Consumption Optimization for Cloud-Based and On-Device Application Uses

3.1 Introduction

In the recent years, smartphones together with the applications running on them evolved rapidly. In 2015, there were more than 7 billion mobile cellular subscriptions worldwide (ICT facts & figures, 2015). As of July 2015, there were 1.6 million apps available for Android users and 1.5 million apps were available in Apple's App Store (Statista, 2015). In our daily lives, we use mobiles to do many activities. The most popular mobile activities performed are social networking, web browsing, emailing, , video watching, and gaming (Mobifroge, 2014). Despite its attractive design, performance, diverse range of capabilities, and multi-functionality and performances; the battery lifetime of smartphone continues to have a significant influence on the user experience. As example, statistics about the consumer satisfaction with battery life of wireless devices in the United States in 2011 show that only 36 percent of the consumers were satisfied with the battery life of their wireless device (Statista, 2011).Another study were conducted in 2013 among 3500 phone users in US, UK, Germany and UAE, showed that battery life is their biggest problem with 1 in 3 (37%) of phone owners(Anera, 2013) .

Battery life has always been a concern not only for smartphone users but also for the manufacturers. Despite the significant efforts to improve the battery technology, the battery capacity could be enhanced only by 5% every year(Robinson, 2009). This fact implies that, the battery technology has not been able to match the demand for power consumption for mobile activities. This constraint of the smartphone batteries results in a limited use of smartphones and emphasizes on battery lifetime extension. Therefore, energy consumption optimization of smartphones has become one of the critical research issues.

Considering the constraint of the smartphone's limited energy, there is no single solution that can fits all users due to the difference in their usage behavior. Therefore, it is necessary to customize the energy consumption optimization to the users. In order to customize the energy consumption to the user we need to define a model to describe the user behavior. Despite the quantitative difference in the usage behavior among users, the model can use the qualitative similarities to describe the behavior of all users (Froehlich et al., 2007).

As users' interest towards saving energy corresponds to their convenience, it is within the users' interest to optimize the energy consumption of his mobile devices to prevent the battery running out of energy on a critical moment (Heikkinen, Nurminen, Smura, & HäMmäInen, 2012). Maximizing their utility is strong intensive for smartphone users to save the energy.

Smartphone usage has been comprehensively studied during the past few years. Some of these studies have focused on investigating the use's interaction with their smartphone (Church et al., 2015; Demumieux & Losquin, 2005; Do, Blom, & Gatica-Perez, 2011; Falaki, Mahajan, et al., 2010; Ferreira et al., 2014; Froehlich et

al., 2007; Jesdabodi & Maalej, 2015; Jones, Ferreira, Hosio, Goncalves, & Kostakos, 2015; Kang et al., 2011; Kim, Ilon, & Altmann, 2013; Lim, Bentley, Kanakam, Ishikawa, & Honiden, 2015; McMillan, Morrison, Brown, Hall, & Chalmers, 2010; Ahmad Rahmati, Tossell, Shepard, Kortum, & Zhong, 2012; Abouzar Rahmati & Zhong, 2013). Other studies have focused on the relationship between the user behavior and the energy consumption of devices (Altmann & Chu, 2001; Altmann & Rohitratana, 2010; Altmann, Rupp, & Varaiya, 1999, 2001; Altmann & Varaiya, 1998; Falaki, Mahajan, et al., 2010; Haile & Altmann, 2013, 2016; Heikkinen et al., 2012; Kang, Park, Seo, Choi, & Hong, 2008; E. Oliver, 2010; Pasricha et al., 2015; N. Ravi, Scott, Han, & Iftode, 2008; Shye et al., 2009; Zhao, Guo, Feng, & Chen, 2011). Many other studies have been devoted to the human-battery interaction (Athukorala et al., 2014; Banerjee, Rahmati, Corner, Rollins, & Zhong, 2007; Ferreira, Dey, & Kostakos, 2011; Ferreira et al., 2013; E. A. Oliver & Keshav, 2011; Ahmad Rahmati et al., 2007; Ahmad Rahmati & Zhong, 2009; Vallina-Rodriguez, Hui, Crowcroft, & Rice, 2010). Their common findings are that all users have a unique usage pattern. Therefore, it is clear that a one-size power management solution cannot fit all and, hence, it is an appropriate solution must adapt to the behavior of the user(Tarkoma et al., 2014). Majority of user studies depends on developing logging application and install it on mobile device to trace the usage behavior. Even though logging techniques can collect large amount of data from a large number of geographically dispersed users, however, they are unable to capture important information such as user intention, and perception; and provide an in-depth understanding of user behavior(Froehlich et al., 2007; McMillan et al., 2010).

Such activity log analysis failed to capture the users' needs and motivations behind their behavior (Church et al., 2015; Froehlich et al., 2007; Heikkinen et al., 2012).

In addition, to enhance the energy efficiency by taking into account the user behavior, it is also believed that, tools for predicting battery life by considering the relationship between power consumption and user activities can help users to extend their mobile battery lifetime. For example, (McCalley, 2002) found that products giving feedback about their energy consumption and having a means to set an energy conservation goal motivate their users to save energy. (Vallerio, 2006) demonstrated how Graphical User Interface (GUI) design can improve energy efficiency of a mobile device. (Abrahamse, 2007) noted that giving consumers more information about their energy consumption resulted in energy savings. This work also contributes to such types of studies assessing the effect of feedback on energy consumption behavior.

There are many available battery awareness applications which automatically increase battery life or help users to manage their smartphone's energy consumption (Creus & Kuulusa, 2007; Oliner, Iyer, Stoica, Lagerspetz, & Tarkoma, 2013; Pathak, Hu, & Zhang, 2012; Peltonen, Lagerspetz, Nurmi, & Tarkoma, 2016; L. Zhang et al., 2010). However, these automated solutions do not typically provide users with a direct indication of the specific actions that make the battery last longer (Athukorala et al., 2014).

Past studies about mobile usage have also explored the user sensitivity towards energy consumption. For example (Heikkinen et al., 2012) conducted a quantitative and qualitative study to understand the behavior and expectation that the

mobile users have regarding energy consumption. Their result shows that users have high sensitivity to energy consumption (i.e. small change in energy consumption has a high impact on user behavior). Also their comparison between user perception and measured values shows that users are reasonably good in evaluating the energy consumption of mobile handset and different services and applications running in them.

Despite the significant efforts by these research works about usage behavior and energy consumption for mobile devices, they did not consider the user preferences in using the applications in terms of the energy consumption. Especially, they did not ask what kind of preferences does a smartphone user has for different types of applications taking into consideration energy consumption aspect.

The main objective of this work is to understand how the user utility can be maximized, subject to the remaining energy in the battery and the user preferences for different types of application uses. In this regard, our work aims at addressing the following questions:

- How the user preferences for different types of applications uses can be modeled taking into consideration energy consumption aspect?
- How the use's utility can be maximized, subject to the remaining energy in the battery and the user preferences for different types of application uses?
- How the model can be applied in the real world?

For the purpose of this study, smartphone application uses were classified into two categories: on-device application uses, which refer to application uses that

do not need Internet connectivity; and cloud-based application uses, which refer to application uses that require Internet connectivity and the cloud. To the best of our knowledge this is the first time this concept of smartphone application usage has been proposed and empirically tested.

To address the research objective, we conduct the following steps: First, we present the design of the model framework. Second, we propose a utility-based energy consumption optimization model to describe the user preferences with respect to energy allocation to the two types of smartphone application uses (i.e., on-device application use and cloud-based application use). The shape of the user utility function is defined by the Cobb-Douglas function. Third, we demonstrate our model by conducting a study with a real smartphone users to explore their preferences for using the smartphone applications through the past usage pattern data regarding energy consumed by the two types of application uses, and through their perceptions for the applications' energy consumption cost by the different types of application uses.

By classifying the smartphone applications based on their used resources we show that the user utility for the energy consumption between different types of applications can be described using simple model like Cobb-Douglass function and the user preferences for allocating the energy can be described using rate of substitution between the two types of the application uses .Our results show the working of our approach and proof that by capturing user preference, an optimal (i.e. utility maximizing) allocation of energy to different types of application uses is possible. The allocation by the user determines not only the ratio between the two types of the application uses but also the quantity with which the energy in the

battery should be consumed. The results also show that, the improvement in the user utility increased with the increased frequency of the user feedback. On the other hand, the results show that if the user has no intervention about energy allocation his utility decreases linearly with respect to energy drain.

The remainder of this chapter is structured as follows. In the next section (section 3.2), we present a literature overview about smartphone user behavior and energy consumption of smartphones. The importance of studying user behavior is presented in section (3.3). Section 3.4 introduces our utility-based energy consumption optimization model. The optimization algorithm is presented in Section 3.5. Section 3.6 presents a case study and application of our model. Section3.7 concludes the chapter with a brief overview, and the research limitations.

3.2 Background

3.2.1 Smartphone Usage

Nowadays, we use smartphones to perform many jobs in our daily lives such as phone calling, Internet browsing, watching videos, downloading files, and sending emails. User behavior has a significant effect on the energy consumption. A significant amount of work has been done on smartphone to understand how users interact with their phones in the real world(Church et al., 2015; Demumieux & Losquin, 2005; Do et al., 2011; Falaki, Mahajan, et al., 2010; Ferreira et al., 2014; Froehlich et al., 2007; Jesdabodi & Maalej, 2015; Jones et al., 2015; Kang et al., 2011; Kim et al., 2013; Lim et al., 2015; McMillan et al., 2010; Ahmad Rahmati et al., 2012; Abouzar Rahmati & Zhong, 2013). A typical methodology that has been

used is to install a logger application on smartphone of the users and collect actual usage data. Their common findings are that all users have a unique device usage pattern. For example, (Kang et al., 2011), who analyze high-level smartphone usage patterns, developed a battery logger for thee Android mobile platform and collected usage log data from 20 smartphone users over a two-month period. Their results showed that all 20 users have their own usage pattern. They also presented a case study, in order to show how to apply usage pattern information to smartphone power management. In their longitudinal study,(Ahmad Rahmati et al., 2012) collected a rich set of data about users including activity logs, and demographic information such as age and household income from 34 university students using iPhone 3GS over one year. Their study highlighted the need to understand the target users of app. They found that social networking apps such as Facebook and YouTube are most popular among users with a lower household income than others. They also found that those users downloaded more apps, used them more frequently, but found them more difficult to use. (Do et al., 2011) conducted a nine month study of Nokia Smartphone usage from 77 users. They collected data about app access, location, and Bluetooth. Their result reveals the dependence of the app usage on the users' location. For example, utility apps such as clocks are used most frequently at home, while camera and map apps are used most frequently on holiday. Participants who spend more time at a friend's home also use communication apps more. Their study highlighted the need for developers to recognize the physical and social usage context of the apps they build. (Lim et al., 2015) conduct one of the largest surveys to date of app users across the world, in order to investigate country differences in mobile app user behavior for 4,824 participants from more than 15 countries

including: USA, China, Japan, Germany, France, Brazil, United Kingdom, Italy, Russia, India, Canada, Spain, Australia, Mexico, and South Korea. Their results show the significant differences in app user behaviors across countries, for example users from USA are more likely to download medical apps, users from the United Kingdom and Canada are more likely to be influenced by price, users from Japan and Australia are less likely to rate apps. Another example is the work of (McMillan et al., 2010), who collected usage data of their iPhone app from a large number of geographically dispersed user (8,676 users) over a period of five months. The authors also supported the activity logs with questionnaires to gain in-depth understanding of user behavior. (Froehlich et al., 2007) developed MyExperience for gathering traces from user's smartphones including device usage such as communication, application usage, and media capture; user context; and environmental sensing, and presented several case studies to demonstrate how it can be used effectively to understand people use and experience mobile technology. For analyzing the smartphone usage differences in user activities. In another study, (Abouzar Rahmati & Zhong, 2013) collected four-month data of HTC wizard phone usage from 14 teenagers in the United States. They found that recreational applications were the most popular, and gaming apps lost their popularity because of the boredom.

In their investigation about whether smartphones induce usage habit,(Jones et al., 2015) present a revisitation analysis of smartphone use. Their analysis three months of application usage from 165 reveals distinct clusters of applications and users which share similar revisitation patterns. However, their results show similarities among much of smartphone usage on a macro-level but, on the other

hand, on a micro-level their results identify unique characteristics in smartphone usage. They argue that “smartphone usage is driven by innate service needs rather than technology characteristics”. Using their three diverse smartphone application usage datasets, (Church et al., 2015) investigate the differences in mobile device usage. They found differences in the top-10 apps in each dataset, in the durations and types of interactions as well as in micro-usage patterns. They argue that it is very challenging to attribute such differences to a specific factor or set of factors. (Jesdabodi & Maalej, 2015) conducted a study on application usage behavior for a period over one year. They segmented usage data into meaningful clusters that correspond to different "states", in which users normally use their smartphone, e.g. socializing or consuming media.

Many user studies have also focused on the relationship between the user behavior and the energy consumption of mobile devices. For instance, (Falaki, Mahajan, et al., 2010) conducted study on smartphone usage patterns using detailed traces from 255 users. They measured the energy consumption and application usage behavior of 33 Android and 222 Windows Mobile devices. Their results revealed the significant variation across users. They make important observations that are relevant to our work. That is, they found that users do not use a large number of different applications (much less than the number of installed applications) but give a large portion of their attention to a small subset of the applications installed. They argued the significance of the personalized application usage models for accurate prediction of battery drain. Their findings strongly motivate the need for customizing smartphone to their users. (Kang et al., 2008) proposed a method to predict discharge behavior of mobile device based on usage patterns. (Heikkinen et

al., 2012) used monitoring traces from multiple years and large number of users supported by questionnaire in order to understand the behavior and expectations of mobile users towards energy consumption. Their study includes both actual behavior of the users and their explicit attitudes, expectations, and experiences. Their results indicate that users need more detailed and clearer information of the battery status and energy consumption of their mobile device. Moreover, users want to understand how different applications and services affect the energy consumption and to learn what they can do to control it. (E. Oliver, 2010) conducted a large-scale smartphone user study to examines how users interact and how users consume energy on their personal mobile devices. (Shye et al., 2009) presented a comprehensive analysis of high-level characteristics of smartphone power consumption. They developed a logger application for Android G1 mobile phones and used it to collect smartphone usage data over a 6-months period from 20 users. Using this data, they applied a linear regression method to build a power model describing the relationship between the power consumption and the user activities. With this model, they also explored optimization potentials.

3.2.2 Human-Battery Interactions

A significant amount of studies have been devoted to the human-battery interaction (Athukorala et al., 2014; Banerjee et al., 2007; Ferreira et al., 2011; Ferreira et al., 2013; E. A. Oliver & Keshav, 2011; Ahmad Rahmati et al., 2007; Ahmad Rahmati & Zhong, 2009; Vallina-Rodriguez et al., 2010). The term human-battery interaction (HBI) was first coined by (Banerjee et al., 2007) to describe mobile phone users' interactions with their cell phones and to manage the battery

energy available. In their user studies on HBI, (Ahmad Rahmati et al., 2007; Ahmad Rahmati & Zhong, 2009) investigated users' attitudes and perceptions of mobile battery life. In particular, they evaluated various aspects of human-battery interactions, including charging behavior, battery indicators, user interfaces for power saving settings, user knowledge, and user reaction. In their experiments, users had to use unfamiliar devices for a period of a month. According to their results, most mobile users had no knowledge of the power characteristics of their devices and applications. Moreover, most mobile users underused the power saving settings of their devices. Although the studies of (Ahmad Rahmati et al., 2007; Ahmad Rahmati & Zhong, 2009) were the first to qualitatively and quantitatively assess how users consumed energy, their studies were conducted in 2006-2007, still before the market launch of the iPhone. Due to the rapid growth of smartphones and their applications, the user behavior (e.g., multimedia playing, gaming) has significantly changed. (Ferreira et al., 2011) presented 4-week study of more than 4000 people to assess their smartphone charging habits to identify timeslots suitable for opportunistic data uploading and power intensive operations on such devices (i.e. when intensive computational operations and long data transfers should be scheduled) as well as opportunities to provide interventions to support better charging behavior. (E. A. Oliver & Keshav, 2011) highlighted the importance of using real user data collected from the world (and how it can influence application development. They collected data from 20,100 Black Berry smartphone users and use it to build an Energy Emulation Toolkit (EET) to allow developers to evaluate the energy consumption requirements of their applications against the collected data. They classify smartphone users based on their charging characteristics. In another

study, (Ferreira et al., 2013) reported on two studies that capture users' experiences with a user-centered battery interface design. In Study 1, they analyzed 12 participants' use of mobile phones and demonstrated that mobile phone users do not know how or what to do to extend their mobile's battery life. In Study 2, they used this information to design, prototype and evaluate an interactive battery interface (IBI) with another 22 participants.

Energy consumption of mobile applications and devices has been investigated from variety of perspectives, including context-aware battery management (N. Ravi et al., 2008), location-aware applications (Anand, 2007), and voice over IP (Gupta and Mohapatra, 2007), and video streaming applications (Xiao, 2008). However, user behavior and attitudes were not directly assessed by these studies. For example, (Ravi et al., 2008) propose a context- aware battery management architecture to predict when the user will charge using user's location traces a (via cell tower) and call -logs. Based on this prediction and the current energy consumption rate, their work determines the battery consumption of applications and warns the user of potential battery depletion. However, these studies did not directly assess user attitudes and behavior.

3.2.3 Battery awareness Applications

In addition to enhancing the energy efficiency by taking into account the user behavior, is also believed that tools for predicting battery life by considering the relationship between power consumption and user activities can help users to extend their mobile battery life (Tarkoma et al., 2014). There are many available

battery awareness applications which automatically increase battery life or help users to manage their smartphone's energy consumption(Creus & Kuulusa, 2007; Oliner et al., 2013; Pathak et al., 2012; Peltonen et al., 2016; L. Zhang et al., 2010), however, these automated solutions do not typically provide users with a direct indication of the specific actions that make the battery last longer (Athukorala et al., 2014). (Athukorala et al., 2014) conducted a study with 1140 users on how the mobile battery-awareness application changes user behavior. The Carat application provides a user with information on the energy consumption of applications. In their study, they classified users into two groups. Users, who had used Carat for more than three months, were considered Advanced Users. Users, who had used it for a shorter time period, were classified as Beginners. Their result showed that advanced users stop using applications that were identified by Carat as highly energy-consuming applications more often than beginners. All users learned to better manage their battery over time.

3.3 The importance of studying user behavior for energy consumption optimization

As smartphone's battery lifetime is dependent on usage behavior, it is necessary to capture statistics on applications usage and their energy consumption to understand and model user preferences. In addition because users may change their usage behavior, especially when the remaining energy in the battery got low, it is also important to understand the motivation behind changing their behavior (Froehlich et al., 2007).

All the studies on usage behavior found that users have their own usage pattern. This diversity among users revealed the need of understanding the user behavior. Better understanding of user behavior helps in multiple ways in developing new mechanisms that better match user expectations (Falaki, Mahajan, et al., 2010; Heikkinen et al., 2012). Technical solutions should learn and adapt user behavior to effectively improve user experience or energy consumption (Falaki, Mahajan, et al., 2010). Without understanding user behavior, it is not possible to clearly understand the impact of any optimization on user experience or the mobile power consumption (Shye et al., 2009).

The majority of user studies depends on developing logging application and installs it on mobile device to trace the usage behavior. Even though logging techniques can collects large amount of data from a large number of geographically dispersed users, however, they are unable to capture important information such as user intention, and perception; and provide an in-depth understanding of user behavior (Froehlich et al., 2007; McMillan et al., 2010). Activity log analysis failed to capture the users' needs and motivations behind their behavior.

3.4 Utility-Based Energy Consumption Optimization Model

In this section, the framework of the user utility –based energy consumption optimization model is presented. After that, smartphone applications were classified into two types of application uses. Then, the modeling of the user preferences for the different types of application uses is explained in details.

3.4.1 Model Framework

In this section, a high level overview of the user utility –based energy consumption optimization model is presented. Figure 2 shows the user utility-based energy consumption optimization framework. The model components and their relations are described below.

- **Battery logger:** The battery logger monitors the energy consumption of the running application. As the user runs the application, the logger runs in the background and collects data about the amount of consumed energy by that application and stores it on the application profile database. It also collects data about the battery level. The various battery loggers were proposed in the literature may be used here. As our work concentrates on the development a user's utility-based energy consumption model, we used one of the existing loggers to trace the energy consumption of users and set the parameters of our model.
- **Network Monitor:** The network monitor works with the battery logger and collects information about the Wi-Fi status (ON/OFF) and status of 3G/4G data communication at the runtime and passes it to the application profile database. Some applications require the WiFi or 3G/4G connection to be active for sending and receiving data. Whereas some applications use the smartphone resources and do not require any data transfer.
- **Application Classifier:** The application classifier takes into account the data collected by the logging tool about the applications which run by the user and the WiFi and 3G/4G connection state. If the connection state of the WiFi or 3G/4G data communication is ON at the runtime of the application, the running application is classified as cloud-based application

uses. But if the connection state of the WiFi or 3G/4G data communication is OFF, the running application is classified as on-device application uses. The application classifier builds a table contains the application and its classification and stores in the application profile database. As the user runs an application the application classifier search the database for the current running application to check if the application exists in the database and is classified. If the application doesn't exist, the application classifier updates the database with the new application and its relevant class.

- **Utility-based Energy Consumption Optimization Algorithm:** This is the core component of our model. It uses the logging data collected by the logging tool about the running applications and their classification to optimize the energy consumption. The optimization algorithm calculates the user utility-maximizing combination of the energy consumption between the different types of application uses based on the past data of the user retrieved from the application profile database. It also takes into account the user feedback about his preference for the energy consumption to readjust the energy allocation each time it received a feedback from the user. The user feedback is received through the user graphical interface.
- **User Graphical Interface (UGI)**: this component displays the status of the energy consumption by the running applications and the optimal allocation of the energy that has to be consumed based on the user utility-based energy consumption algorithm. It also enables the user to switch the energy allocation between the different types of application uses. The user graphical interface also provides the user with notifications asking him for a feedback

about his preferences for the energy allocation. The notifications can also be warning messages that can help the user to control his usage behavior

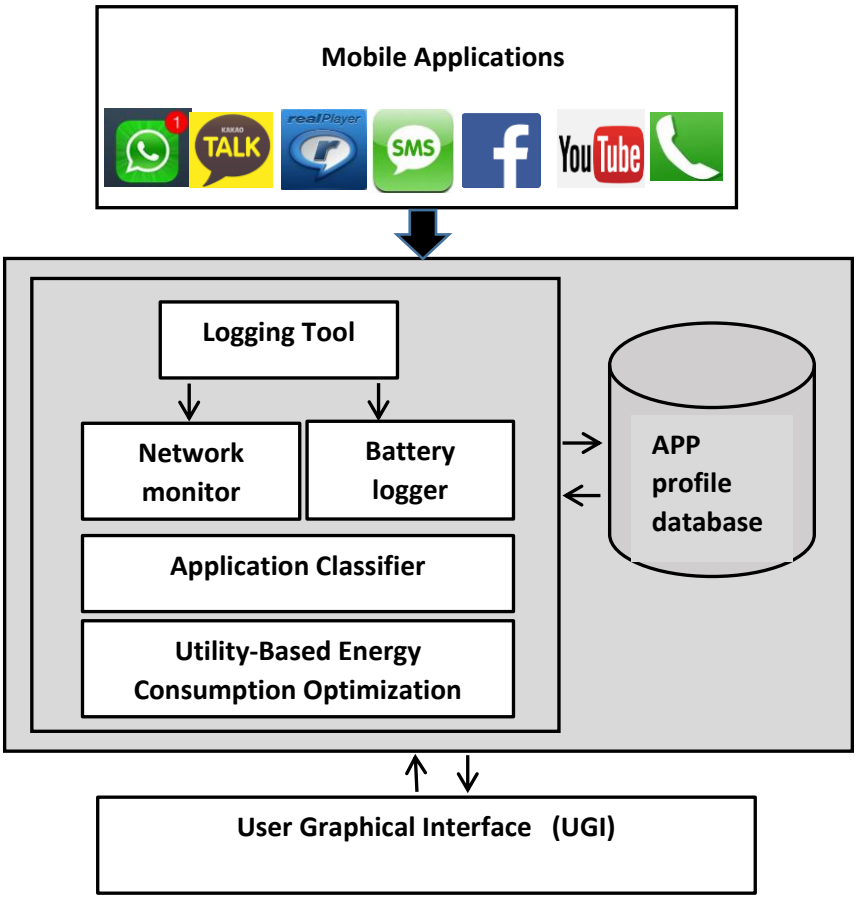


Figure 3. Model Framework

.

- **App profile Database:** Stores the information collected by the logging tool about the applications usage and their energy consumption as well as their classification made by the application classifier. It also makes use of the energy consumption data that is optimized by the application algorithm to update its database.

3.4.2 Classification of Application Uses on Smartphones

With the rapid growth of smartphone applications, smartphone have become rich in the number of applications installed. Some applications (e.g., video streaming and gaming (Falaki, Lymberopoulos, Mahajan, Kandula, & Estrin, 2010) are very much resource intensive in terms of processing power and data transfer (Carroll & Heiser, 2010). They are highly energy-consuming applications and, consequently, drain smartphone batteries very quickly.

Despite the diverse range of activities that can be performed on the smartphone, the most popular mobile activities performed are social networking, web browsing, emailing, video watching, and gaming. These activities can be performed either locally on-device or through cloud-based services. For example, if a smartphone user wants to send a message, the user can perform the activity online using one of the social networking app such (Whatsapp, Viber, Kakaotlak, etc) or the user can perform it off-line and send sms. Also if he wants to make a phone call, he can use the cellular network or he can use VOIP. Another example is that a user who wants to watch a video, the user can perform the activity online using Youtube or he can download it into his smartphone and watch it latter. Applications such as phone calling, sms use the phone network, without connecting to the internet, and are considered on-device applications. Whereas VOIP, social networking services, and Youtube are cloud-based applications. Therefore, we classify these smartphone application uses into two categories: on-device application uses, referring to applications that can be executed on the smartphone locally; and cloud-based application uses, referring to applications, which require data transfer and processing

power on the cloud. We consider these two application use categories to differ in their utility to the user, depending on the energy stored in the battery.

3.4.3 Modeling the User Preferences for the Different Types of Application uses

In spite of the large number of applications installed on smartphones, users do not use them equally. In fact, as users have different preferences for their application uses, their valuation of the amount of energy consumed per type of application use is different. Some users may value playing online games very highly, while others are content playing local games. Within this paper, we consider the preferences for two types of application uses (section 3.4.2), namely on-device application uses and cloud-based application uses.

For smartphone users, the trade-off between using an application on-device or cloud-based is significant. That is, both alternatives can be thought of as representing a bundle of different characteristics: execution time, energy consumption, and monetary cost. For example, considering the battery constraint of the smartphone, the user might wish to maximize her utility from the amount of energy remaining on her smartphone, as an opportunity for charging the smartphone does not exist in the near future. However, the amount of energy he can consume is constrained by the amount of the remaining energy in the battery as well as the user's preference for each type of application. To express the user preference for each type of application use, it is necessary to apply the theory of consumer behavior (Allen, 2005) and define the user's utility function.

With respect to the utility function, the Cobb-Douglas is useful, as it is well-behaved (i.e., more of a good is preferred to less of a good (Allen, 2005). Cobb-

Douglas functions are widely used in economics to represent the preferences between goods and the utility that can be obtained by those goods (Brenes, 2011; Yan, Shi, & Yu, 2013). However, some other works applied the Cobb-Douglas in other domains. For example, (Hasan, Kamil, Mustafa, & Baten, 2012) used a Cobb-Douglas stochastic frontier model to measure the domestic bank efficiency in Malaysia. (Hayes, 1983) applies the Cobb-Douglas model to the association of research libraries.

In this work, the Cobb-Douglas utility function describes the user preference with respect to energy allocation to the two types of application uses (i.e. on-device application use and cloud-based application use). Consequently, it defines combinations of the energy consumption by these two types of application uses which result in the same utility for the user. That means, if one combination has more energy consumption by on type of application uses than a second combination, it must have less energy consumption for the other type of application uses. Therefore, our user utility function is represented by the following Cobb-Douglas function:

$$U(E_{local}, E_{cloud}) = (E_{local})^{\alpha} (E_{cloud})^{\beta} \qquad (3\text{-}1)$$

where E_{local} is the energy planned to be consumed through on-device application uses, E_{cloud} is the energy planned to be consumed through cloud-based application uses. The unit of E_{local} and E_{cloud} is [min].

The exponents, α and β, of the Cobb-Douglas utility function should be defined, such that they express the preferences of the user for the two types of application uses: cloud-based application uses and on-device application uses. The user preferences for using the smartphone applications can be captured through the

past usage pattern data regarding energy consumed by the two types of application uses and also through their perceived costs for allocating energy for the different types of application uses. For our model, we consider the past consumption of energy for on-device application uses $\mathrm{E}^{\mathrm{H}}_{\mathrm{local}}$ and the past consumption of energy for cloud-based application uses $\mathrm{E}^{\mathrm{H}}_{\mathrm{cloud}}$. The past consumption of energy can be obtained through monitoring applications, running on the smartphone of the user. Using the monitored energy consumption for *n* battery usage periods, the average amount of battery energy consumed through on-device application uses $\mathrm{E}^{\mathrm{H}}_{\mathrm{local}}$ and cloud-based application uses $\mathrm{E}^{\mathrm{H}}_{\mathrm{cloud}}$ for each user can be calculated as follows:

$$E^{H}_{local} = \frac{\sum_{k=1}^{n} E^{Hk}_{local}}{n} \qquad (3-2)$$

And,

$$E^{H}_{cloud} = \frac{\sum_{k=1}^{n} E^{Hk}_{cloud}}{n} \qquad (3-3)$$

Where E^{Hk}_{local} is the amount of energy consumed by on-device application uses at usage period *k* , E^{Hk}_{cloud} is the amount of energy consumed by cloud-based application uses at usage period *k* , and *n* is the total number of usage periods. The relative ratio of $\mathrm{E}^{\mathrm{H}}_{\mathrm{local}}$ and $\mathrm{E}^{\mathrm{H}}_{\mathrm{cloud}}$ is then used to set α and β as shown in the following two equations:

$$\alpha = \frac{\mathrm{E}^{\mathrm{H}}_{\mathrm{local}}}{\mathrm{E}^{\mathrm{H}}_{\mathrm{local}} + \mathrm{E}^{\mathrm{H}}_{\mathrm{cloud}}} \qquad (3\text{-}4)$$

$$\beta = \frac{\mathrm{E}^{\mathrm{H}}_{\mathrm{cloud}}}{\mathrm{E}^{\mathrm{H}}_{\mathrm{local}} + \mathrm{E}^{\mathrm{H}}_{\mathrm{cloud}}} \qquad (3\text{-}5)$$

According to the definition of Eq 3-4 and Eq 3- 5, the factors α and β are positive and fulfil α+β=1.

To calculate the optimal allocation of energy E^*_{local} and E^*_{cloud} to the two types of application uses, we also need to consider the budget constraint, which is expressed with the following equation:

$$E_{local} * P^E_{local} + E_{cloud} * P^E_{cloud} = m \qquad (3\text{-}6)$$

Where P^E_{local} is the user's perceived cost for the energy consumption by on-device application uses, P^E_{cloud} is user's perceived cost for the energy consumption by cloud-based application uses, and *m* represents the amount of the energy that is left in the battery (referred to as energy budget) .

To be able to use Eq 3-6, our first step towards gaining understanding the energy budget *m*, was to convert the scaling of the energy remaining in the smartphone battery into monetary values in $. For example, if the battery energy level is 100%, we assume that the value of m is 100$. Similarly, if the battery energy level is 90%, we assume that the value of m is 90$, and so on.

With respect to the perceived cost for the energy consumption, a user is asked to express her preferences for consuming energy for on-device application uses and for cloud-based application uses at different values of *m*. The user's preference for using an application type can be expressed by stating the perceived cost (P^E_{local} and P^E_{cloud} [$/min]) for the energy for both types of application uses. In this regard, we would like to note that user's perceived cost refers to the user's valuation for energy planned to be consumed by on-device application uses or

cloud-based application uses but not to the real prices of energy. The point is that, different users may have different perceived costs; therefore, prices measure the rate at which users are just willing to substitute the energy consumption of one type of application for another. As prices are not arbitrary numbers (Varian, 2014) but reflect how user values the energy consumption, it is sufficient to state the rate of substitution RS for the cost of the two types of application uses as follows:

$$\mathrm{RS} = \frac{\mathrm{P}^{\mathrm{E}}_{\mathrm{local}}}{\mathrm{P}^{\mathrm{E}}_{\mathrm{cloud}}} \qquad (3-7)$$

For example, if the perceived cost of energy consumption by on-device application uses is 2$/min and the perceived cost of energy consumption by cloud-based application uses is 1$/min, then the rate of substitution for this user is 2. That means, the user is willing to give up 2 unit [min] of energy for cloud-based application uses only if the user obtains 1 units [min] of energy for on-device application uses. In this regard, if we observe one energy consumption pattern for one complete usage period we get RS for one usage. If the usage behavior changes we observe consumption pattern and we get another RS. As we observe more and more energy consumption pattern we learn more and more about the shape of the underlying preferences that may have generated the observed usage behavior.

With input values for $\mathrm{E}^{\mathrm{H}}_{\mathrm{local}}$, $\mathrm{E}^{\mathrm{H}}_{\mathrm{cloud}}$, $\mathrm{P}^{\mathrm{E}}_{\mathrm{local}}$, $\mathrm{P}^{\mathrm{E}}_{\mathrm{cloud}}$, and *m*, the utility-maximizing allocation to on-device application uses $\mathrm{E}^{*}_{\mathrm{local}}$ and cloud-based application uses $\mathrm{E}^{*}_{\mathrm{cloud}}$ can be calculated by using the Lagrange multipliers theorem (Varian, 2014). The results provide the optimal amounts of energy that should be consumed through on-device application uses and through cloud-based application

uses (i.e., the amount of units [min]). In detail, the optimal combination can be calculated as follows:

$$E^*_{local} = \frac{\alpha}{\alpha+\beta}\frac{m}{P^E_{local}} \tag{3-8}$$

$$E^*_{cloud} = \frac{\beta}{\alpha+\beta}\frac{m}{P^E_{cloud}} \tag{3-9}$$

3.5 Optimization Algorithm

The utility-maximizing energy consumption optimization for the on-device application uses and cloud-based application uses requires adjusting the energy allocation between the applications based on their usage preferences by the user. Considering the different applications belong to each class of the application uses, the applications need to be ranked based on their amount of energy consumption. Initially, the values of the energy consumption for on-device application uses and cloud-based application uses for the usage period *k* can be set using the values of the energy consumption by the past usage period *k-1*, or by the average values of the past energy consumption of *n* usage periods. These initial values define the planned combination of energy consumption allocation between the two types of the application uses. Considering that the user has no opportunity to charge his smartphone, the user preferences for allocating the energy between the applications may change as the level of the remaining energy goes down. Therefore, tracing the user preferences at different energy level and getting feedback about his perceptions of the applications' energy consumption cost can help to readjust the energy allocation between the two types of the application uses so that the user utility will

be maximized. The algorithm to optimize the energy allocation between the two types of the application uses is briefly illustrated in Figure 4 below:

(0) Ranked list of running local apps and ranked list of running cloud apps and their energy consumption

(1) For each usage period k, set E^{H}_{local}, E^{H}_{cloud} of the usage period k-1 (or the average values of n periods from the past) as the initial values for the energy planned to be consumed by the on-device application uses, and cloud-based application uses respectively.

(2) While Applications are running and electricity cable is not plugged in do

(2.1) Ask user for RS (P^{E}_{local} and P^{E}_{cloud})

(2.2) Calculate E^{*}_{local}, E^{*}_{cloud} and the corresponding utility.

(2.3) Adjust the energy allocation to the local and cloud applications such that they fulfill E^{*}_{local} and E^{*}_{cloud}

(2.4) Update E^{H}_{local}, and E^{H}_{cloud} to be E^{H*}_{local} and E^{H*}_{cloud} respectively

(2.5) Calculate the new values of α and β, E^{*}_{local}, E^{*}_{cloud} and corresponding utility.

(3) While end

Figure 4. Optimization Algorithm

3.6 Simulation

3.6.1 Data Collection

As mentioned in the introduction section, for the purpose of our simulation, we collected data about users' usage behavior with two approaches: usage monitoring using logging tool, and questionnaires. We choose Android OS based smartphone as it is popular and it has the biggest market share in the smartphone industry. According to data from the International Data Corporation (IDC) Worldwide Quarterly Mobile Phone Tracker, Android dominated the market with an 82.8% share in 2015 Q2 (IDC, 2015). We recruited 10 participants all of them are males. Nine of the participants were between 24-34 years old, while one of them is 45 years old. Participants were recruited from different cities in South Korea. In general, Internet penetration rate in Korea is very high and WiFi networks are deployed widely. All participants are students in different majors including engineering, medicine, and business administration. Their education levels ranged from bachelor's degree to Ph.D. All participants were familiar with using smartphones and active users of smartphone applications like social network apps.

Data were collected by asking the participants to install *eStar Energy Saver* application on their own smartphones .The application is available for free on play store of Android smartphones. It is easy to access and it provides information about the percentage of energy consumed by the applications launched by the user as well as system applications. It also shows the time duration for each application usage. As our simulation based on the usage behavior/preference, we only focus on the applications launched by the user.

The participants were asked to send their battery consumption data monitored by the *eStar Energy Saver* application for each compete battery usage period. Battery usage period refers to one complete battery usage without connecting the mobile to the charger (one complete discharging period). It begins when the smartphone battery is full, and ends when the smartphone battery is discharged. We collected 10 complete usage periods for each user.

3.6.1.1 Applications

Based on the data of past behaviors that have been obtained by monitoring the users' application uses, we found the most common applications among all users for each class. For example, in case of on-device application uses, the most common applications are: phone calling, sms, camera, gallery, offline games, video player , and music player. On the other hand, the most common cloud-based applications are: Facebook, Facebook Messenger, Whatsapp, KakaoTalk, Viber, Line, Youtube, Email, Internet browser, Instegram, Skype, Twitter, Translate, and play Store.

Figure 5 presents the application used by each user in terms of the amount of the consumed energy. To conduct this analysis we calculate the average energy consumed by each application for the monitoring period for each user. To do that, firstly, for each application, we calculated the energy consumed by each application for each usage period. Using the obtained values for 10 battery usage periods we then calculated the average amount of battery energy consumed by each application for each user. The applications were ranked based on their average consumed energy for the monitored usage periods.

The result of Figure 5 shows that, Facebook, Messenger, KakaoTalk, WhatsApp, YoutTube, email, and internet browser are the most popular applications in the cloud-based applications category among most of the users. This is mainly due to the fact that, recently these applications become more popular than other applications. On the other hand, the results show that video player, music , and phone calling are the most popular on-device applicant uses. Another on-device application use is Games app which appeared popular among User4, User7, User 9, and User 10. Interestingly, Games app appears as the top application for the User 10. Overall, we find that, the amount of energy consumed by each application differs across the users. That is, the relative consumed energy by each application is mainly depends on the preferences of each user.

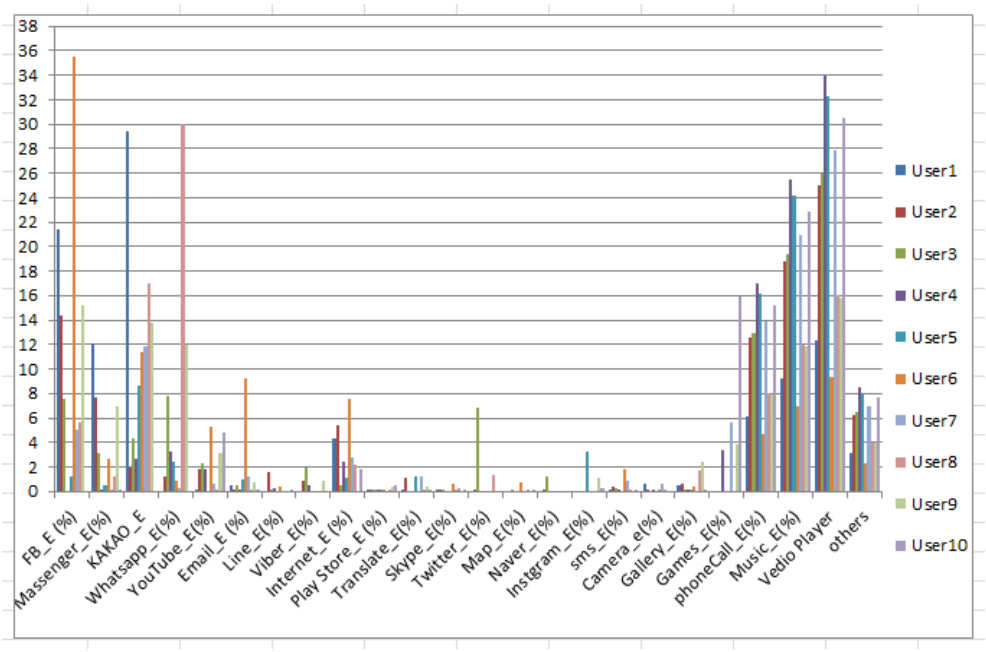


Figure 5. Average energy consumed by each application for each user

3.6.1.2 Energy Relation

As mentioned, in spite of the large number of applications installed, users do not use them equally. Our results confirmed that users have different preferences for their application uses. Moreover, we find that, for each user the amount of the consumed energy by each application differs across the usage periods. This variation is related to the amount of time spent by the user on that application for each period. Figure 6 shows the top-10 applications for a sample user (User 1) in terms of the consumed energy. From the figure, we can see the differences in the top-10 apps in each usage period. The figure shows the energy consumed by each application for each monitoring period. We find that, All the top -6 applications appear in the all periods with a significant amount of consumed energy. With the exception of Facebook which didn't appear in period2. Those applications are: KakaoTalk, Facebook, Video player, Messenger, Music, and phone call. Overall, the analysis shows that User 1 is highly prefers social applications. Therefore, he should consider how to allocate the energy in order to keep in contact as longer as possible.

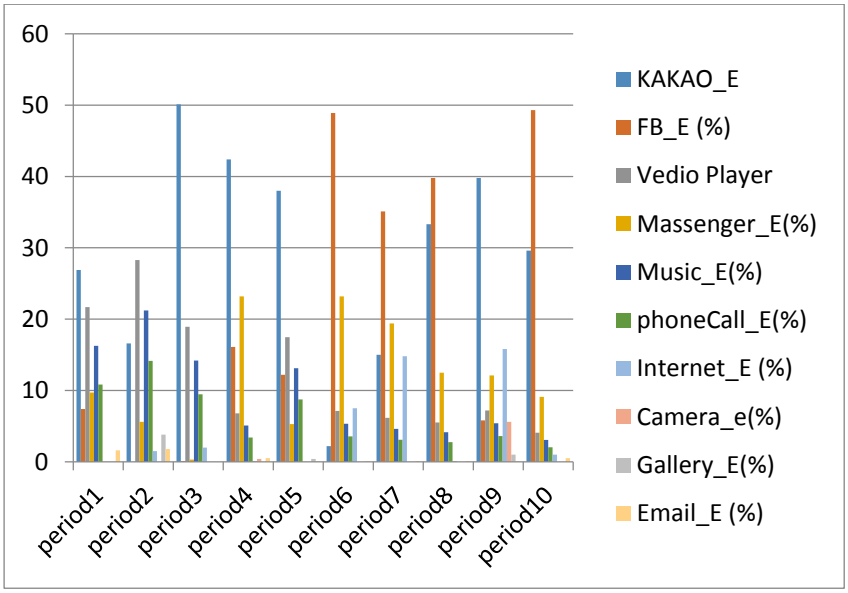


Figure 6. The top-10 applications for a sample user (User 1) in terms of the consumed energy for each usage period.

3.6.1.3 The Long-term preferences for the energy consumption

Based on obtained values, the long-term preferences for the energy consumption for each user is modeled using the utility function (Eq 3- 1) with α (Eq 3-4) and β (Eq 3-5) based on the two types of application uses. Figure 7 shows the long-term preferences for the on-device application uses and cloud-based application uses for the 10 users according to their average energy consumption through the two types of applications uses during the monitored energy usage periods.

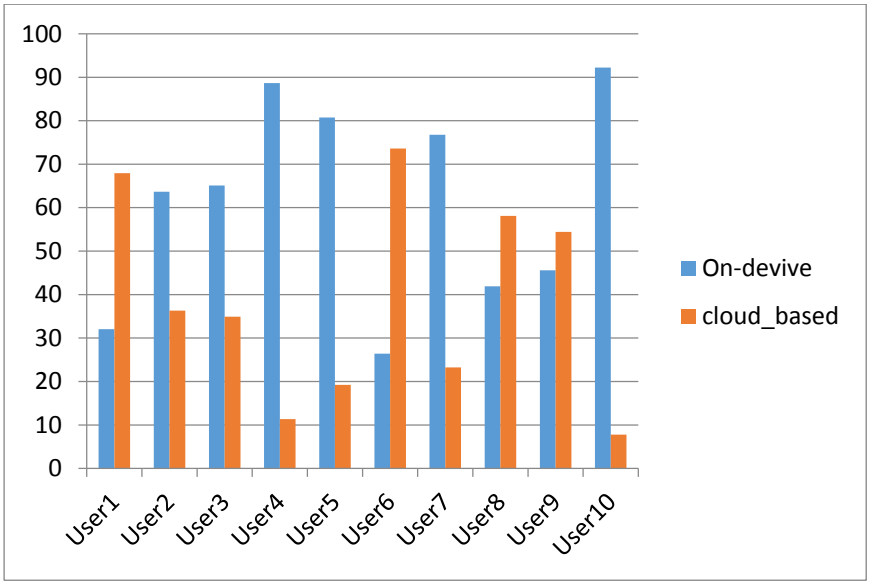


Figure 7. The long-term preferences based on the past energy consumption through on-device application uses, and cloud-based application uses for the 10 users

Figure 7 shows that the energy consumption preferences of Users1, User6, User8, and User9 for cloud-based application uses are higher than their preferences for on-device application uses. Whereas other users' energy consumption preferences for on-device application uses are higher than cloud-based application uses. For example, for User1, the ratio of the on-device application uses and cloud-based application uses is about 1:2. That means that User 1's preference for on-device application uses is half as low as his preference for cloud-based application uses. In case of User 3, the ratio of the on-device application uses and cloud-based application tends to be 2:1. That means that User 3's preference for on-device application uses is twice as high as his preference for cloud-based application uses. Based on these data, we grouped the participants into two classes: cloud-based application uses higher preference users (User1, User6, User8, , and User9), and on-

device application uses higher preference users (User2, User3, User4, User5, User7, and User 10).

Based on the values of the energy consumed by the top-10 applications for each usage period, the following table (Table2) summarizes the maximum and minimum energy consumption rate for all of the 10-usage periods for a sample user (User1). The zero value of the minimum consumed energy of some applications , particularly the last four of the top -10, indicates that , these applications were never used by the user in that usage period.

Table 2. Maximum and minimum energy consumption rate for all of the 10-usage periods for a sample user (User1).

Application	Maximum consumed energy (%)	Minimum Consumed Energy (%)
KAKAO_E	50.1	2.2
Facebook	49.3	0
Vedio Player	28.28	4.08
FB_Massenger	23.2	0.3
Music player	21.21	3.06
phoneCall_	14.14	2.04
Internet	15.8	0
Camera_	5.6	0
Gallery	3.8	0
Email	1.8	0

3.6.1.4 User-Behavior Traces Over Time and Different Energy Levels

After collecting their usage data through the monitoring application, qualitative data regarding users' perceptions of the applications' energy consumption cost were collected via in-person interviews. Combining the questionnaires with the monitoring studies can help to see whether the user's action matches their usage behavior (Heikkinen et al., 2012) and to provide in-depth

understanding about the users' needs and rationale behind their behavior (McMillan et al., 2010).

We discussed the participants about their perceptions of mobile energy consumption. We asked them how they prefer to spend the battery energy for the different applications uses and how they change their usage behavior according the energy remaining in their battery. In details, we asked the participants to express their perceived costs for allocating energy to on-device application uses and to cloud-based application uses (P^{E}_{local} and P^{E}_{cloud} [$/min] respectively) at different levels of energy remained in the battery. Our interviews emphasized the importance of user preference for the energy consumption. Not surprisingly; all participants have different perceptions for the energy consumption and change their usage behavior as the level of the battery energy goes down based on their applications preferences. Participants indicated that, their perceived cost is varied according to the energy remaining in their smartphone battery. They noted that if the level of the remained energy in the battery goes low they tend to give up some amount of energy consumption by one type of application uses for the other preferred type of application uses. This analysis led us to emphasize the importance of adopting the rate of substitution (RS) concept as a measure to explore the utility maximization combination of energy allocation to on-device application uses E^{*}_{local} and to cloud-based application uses E^{*}_{cloud} . As mentioned, in this study, the rate of substitution defines how many units of energy the user would be willing to sacrifice of one type of the application uses to consume more units of energy for the other type of the application uses. The RS is value is calculated using Eq 3-7. Table 3 presents the

perceived cost for the energy consumption at different energy levels of sample users (User1, User6 and User 8).

Table 3. perceived cost for the energy consumption at different energy levels

m	User1		User6		User8	
	P^E_{local}	P^E_{cloud}	P^E_{local}	P^E_{cloud}	P^E_{local}	P^E_{cloud}
100	2	1	2	1	1	1
90	2	1	2	1	1	1
80	2	1	2	1	1	1
70	2	1	2	1	1	1
60	2	1	4	1	1	1
50	2	1	4	1	1	1
40	4	1	4	1	2	1
30	4	1	4	1	2	1
20	4	1	4	1	2	1
10	4	1	4	1	2	1

As shown in the table 3, at the high level of the remained energy in the battery (100% ~ 50%) the RS value of the user1 is 2. Which means that to get one more minute for the energy to be consumed by the on-device application uses, user 1 must give up 2 minutes of energy for cloud-based application uses. Also, the table shows that, when the energy level goes lower than 50% the RS values become 4. Which means that, in order to consume one more minute of energy by the on-device application uses, user 1 must give up 4 minutes of energy for cloud-based application uses. This behavior of the user1 is in line with the long-term preferences of user 1 for the cloud-based application uses that already shown in figure 7. Therefore, the high cost for the energy consumption by the on-device application uses, the less preferences to consume battery energy by the on-device application

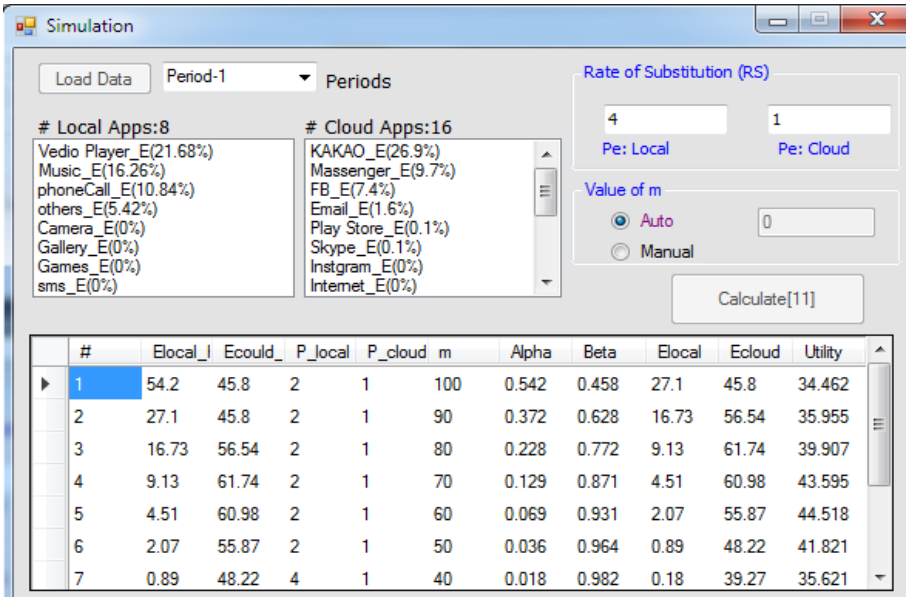
uses is to the user 1. User 6 almost has the same behavior as he also has higher preferences for the cloud-based application uses. The only difference is that the RS values are changed at 60% and below of the remained energy. In case of user 8, although his long-term preferences for the energy consumption show that he has higher preferences for the cloud-based application uses. However, his behavior at the high levels of the remained energy tends to be neutral. That is, the value of RS is 1. Only when the battery energy level goes below the 50% the value of the RS changes to 2.

3.6.2. Simulation Results

3.6.2.1 The Improvement in utility

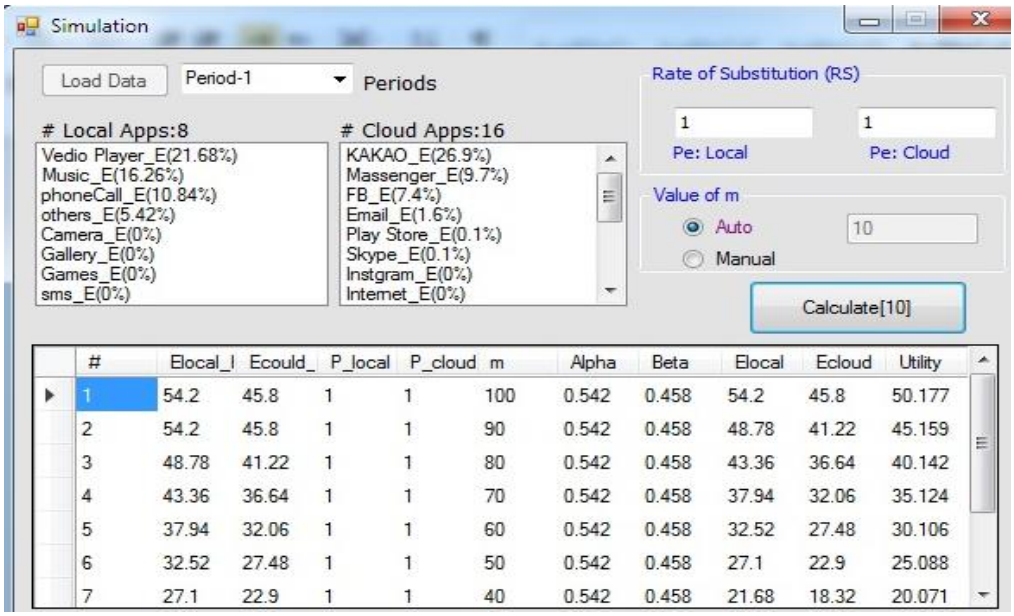
Following the steps of the optimization algorithm presented in section 3.5, the simulation results in this section show the effect of the users' perceived costs for the energy allocation to on-device application uses and cloud-based application uses on the user's utility for a sample user (User1). The simulation is conducted using the C# programming. Figure 8 shows the utility-maximizing combination of energy allocation to the two types of the application uses and the corresponding utility at different levels of smartphone's battery remained energy. As shown in the figure, at the first half of the energy remained in the battery (from level 100% till 50 %), the user perceived costs for allocating the energy to the on-device application uses and cloud-based application uses are 2 $ /min and 1$/min respectively (RS=2). The corresponding utility is increased as the energy level goes down except at level 50% where the utility value decreased. However, as the battery levels goes down below

50% the user perceived costs, the user perceived costs for allocating the energy to the on-device application uses and cloud-based application uses become 4 $ /min and 1$/min respectively (RS=4). The corresponding utility is decreased.



#	Elocal_l	Ecould_	P_local	P_cloud	m	Alpha	Beta	Elocal	Ecloud	Utility
1	54.2	45.8	2	1	100	0.542	0.458	27.1	45.8	34.462
2	27.1	45.8	2	1	90	0.372	0.628	16.73	56.54	35.955
3	16.73	56.54	2	1	80	0.228	0.772	9.13	61.74	39.907
4	9.13	61.74	2	1	70	0.129	0.871	4.51	60.98	43.595
5	4.51	60.98	2	1	60	0.069	0.931	2.07	55.87	44.518
6	2.07	55.87	2	1	50	0.036	0.964	0.89	48.22	41.821
7	0.89	48.22	4	1	40	0.018	0.982	0.18	39.27	35.621

Figure 8. Simulation for a sample user (User 1) considering his feedback (variable values of RS)



#	Elocal_l	Ecould_	P_local	P_cloud	m	Alpha	Beta	Elocal	Ecloud	Utility
1	54.2	45.8	1	1	100	0.542	0.458	54.2	45.8	50.177
2	54.2	45.8	1	1	90	0.542	0.458	48.78	41.22	45.159
3	48.78	41.22	1	1	80	0.542	0.458	43.36	36.64	40.142
4	43.36	36.64	1	1	70	0.542	0.458	37.94	32.06	35.124
5	37.94	32.06	1	1	60	0.542	0.458	32.52	27.48	30.106
6	32.52	27.48	1	1	50	0.542	0.458	27.1	22.9	25.088
7	27.1	22.9	1	1	40	0.542	0.458	21.68	18.32	20.071

Figure 9. Simulation for a sample user (User 1) without his feedback (constant values of RS)

For assessing the advantage of the proposed optimization model for the improvement in the utility, the measured utility obtained in the simulation results based on the user feedback about his preferences for allocating the energy between the two types of application uses at different energy levels is compared against a case with constant values of RS , α and β (without any feedback from the user).

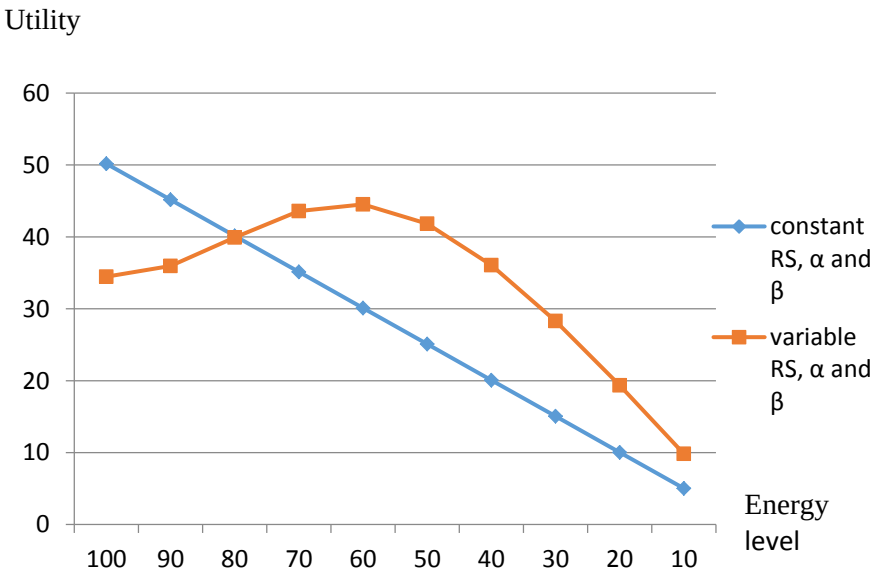


Figure 10. Effect of the user feedback on the utility

The user's utility for the case of constant values of RS, α and β is calculated to obtain baseline figure for the change in the user's utility at different levels of energy remained in the battery. In his case The values of α and β are 0.542 and 0.458 respectively (as presented in Figure 9).

As shown in Figure 10, the user's utility for the case that doesn't depend on the user feedback about his preference for allocating the energy between the two types of the application uses steady deceases at fixed rate as the remained energy

level goes down. It decreases linearly with respect to energy drain. The constant decreasing is due to the constant values of RS, α and β. The figure also shows that, at high levels of the battery energy (100% ~ 90%) the user's utility of the case that depends on the user feedback is lower than the case that doesn't depends on the user feedback. However, the user utility increase as the user provides feedback about his preferences for allocating the energy between the two types of the application uses. Whereas for the second case, as the battery decreases the user's utility keeps decreasing at fixed rate as the user has no preferences for the energy allocation.

Although the high values of the user's perceived cost for allocating the energy between the on-device application uses and cloud-based application uses (with RS=4) at the second half of the remained energy of in the battery (lower than 50%) have resulted in decreasing the user's utility , however, the user's utility still higher than those of the case that doesn't have any feedback from the user. For example, at very low energy level, the user's utility for the case without user feedback (fixed values of RS, α and β) equals to 5 whereas for the case with user feedback (variable values of RS , α and β) is around 10 which is double. Overall, the results show that the user's utility can be improved with getting feedback from the user about his preferences for allocation the energy between the different types of application uses.

3.6.2.2 The Impact of the frequency of user feedback on the utility

As shown the previous section, one of the advantages of user feedback about his preferences for the energy allocation between the different types of application uses is that, it improves the user utility. It helps in readjusting the energy

consumption between those application uses based on the user preferences. The benefit of using the user feedback can be examined by its frequency. In order to analyze the impact of the frequency of user feedback on the utility, we trace the user feedback at different ranges of the battery energy levels using the data of a sample user (User1).

Figure 11 presents the change in the utility based on the frequency of the user feedback of four cases: first case is receiving the user feedback at range of 10% difference from the previous user feedback. In another words, receiving the user feedback as the battery level has gone down by 10%.That implies a frequency of 10 user's feedback. Second case is receiving the user feedback at range of 20% (a frequency of 5 user's feedback). Third case represents receiving the user feedback at range of 40% (a frequency of 3 user's feedback), and fourth case is receiving the user feedback after the energy level has gone down by 50% (a range of 50%) which implies a frequency of only 2 user's feedback.

The results show that, increased number of frequency of user feedback results in higher utility. For instance, for the case of receiving the user feedback at 10% range from the previous feedback, the corresponding utility values at every level are higher than those of all the other cases which have less frequency of user feedback. For example, at energy level of 20%, the utility value for the user feedback of range 10% is 19.36, whereas the utility value for the user feedback of range 20% is 16. 3 and it is only 8.517 for the user feedback of 40%. Another example, receiving the user feedback after every 10% decreasing in the battery energy level results in user utility of 41.821 which is double than the user utility of receiving the user feedback every 50% decreasing in the battery energy level (only

19.975). Similarly the user utility values of the case of user feedback of range of 20% are higher than the user utility values of the cases of user feedback of ranges 40% and 50%. And the user utility values of the case of user feedback of range of 40% are higher than the user utility value of the cases of user feedback 50%.

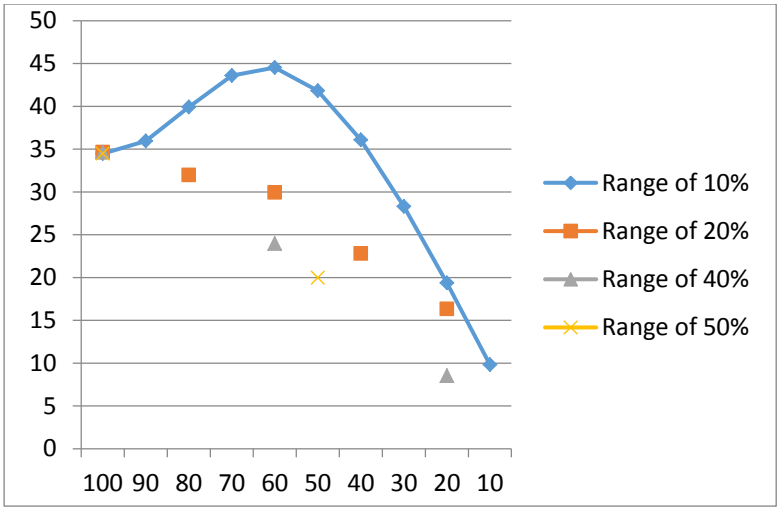


Figure 11. Impact of the frequency of user feedback on the utility

Overall, the results of the different cases of the frequency of user feedback indicate that the user utility is increased as the number of the user feedback about his preference in energy allocation between the different types of application uses increased. That increased value in the user utility is related to the optimal amount of energy determined by the optimization algorithm after each user feedback. That is, the optimization model helps the user to adjust his energy consumption between the different types of the application uses such that it maximizes his utility according to his preferences.

3.7 Discussion and Conclusion

In this chapter, we proposed a utility-based energy consumption optimization model, which considers the user's preferences with respect to energy consumption aspects when using smartphone applications. The smartphone applications were classified based on their use (on-device application uses and cloud-based application uses). A utility-based energy consumption optimization algorithm model was proposed taking into account the user preferences to provide optimal allocation of energy between those different types of application uses. By classifying the smartphone applications based on their used resources we show that the user utility for the energy consumption between different types of applications can be described using simple model like Cobb-Douglass function and the user preferences for allocating the energy can be described using rate of substitution between the two types of the application uses. The model implementation was presented and its performance was evaluated through a case study in which the users' preferences were reflected through the past patterns of energy consumption, and their feedback about their perceived costs for allocating energy to each type of application uses. Using the collected data, we applied our proposed model to analyze the improvement in their utility and the impact of the frequency of user feedback on the user utility. The results show that the model can help optimize the energy consumption in such a way that improves the user utility. The user only needs to specify the perceived cost of energy for each type of application. The optimization determines not only the ratio between the two types of application uses but also the quantity with which the energy in the battery should

be consumed. The results also show that, the improvement in the user utility increased with the increased frequency of the user feedback. On the other hand, the results show that if the user has no intervention about energy allocation his utility decreases linearly with respect to energy drain.

We argue that this work provides some additional interesting understanding of smartphone usage behavior, and it will contribute to the line of studies assessing the effect of usage behavior on energy consumption optimization of smartphones. It can be used by smartphone users to obtain the maximum utility from the remaining energy in their smartphones battery. That is, a smartphone user with a Cobb-Douglas utility function can maximize her utility by allocating a fraction of her remaining energy to each type of application uses. The size of the fraction is determined by the user's preference, and her perceived cost for the energy consumption of both types.

The results of this study can also be used by the smartphone manufactures and developers to enhance the smartphone's platform. They should recognize that there is a need to implement a learning mechanism to monitor user behavior and learn his preference and provide the user with a feedback that could help him to adjust his usage behavior and take proactive measures to allocate the energy consumption between the different types of the applications uses.

Although the proposed model does not completely describe the behavior of the user, it captures very significant factors of user preferences and represents a first step towards understanding and complete modeling of the user preferences for the different types of application with respect to energy consumption.

Chapter 4. User-Preference-Based Optimization for Making Multi-Criteria Offloading Decisions in Mobile Clouds

4.1 Introduction

Smartphones increasingly became an essential part of human life as the most effective and convenient communication tools not bounded by time and place(Dinh, Lee, Niyato, & Wang, 2013) . In contrast to their capabilities and functionalities (i.e., their small size, light weight, attractive design, and their provision of useful services), smartphones have limited resources, such as battery energy, processing power, network bandwidth, and storage capacity (Altamimi, Palit, Naik, & Nayak, 2012; Kumar et al., 2013). These constrains of mobile resources boost the need for mobile cloud computing (MCC).

MCC has recently been introduced to overcome the limitations of smartphones by enabling mobile devices to offload the computational intensive applications on resourceful clouds instead of running them locally on the device. This is referred to as computation offloading(Enzai et al., 2014). However, transferring the execution of a mobile application to the cloud requires not only communication between the mobile device and the cloud but definitely introduces costs for the mobile user, based on the exact resources consumed (X. Zhang et al., 2011). It might consume additional energy, bandwidth, and money to cover the cost for communication and cloud computations. Therefore, if the offloading cost could

exceed the cost of local execution as a result of the additional resource utilization in the offloading process(Kosta et al., 2012), it might not be desirable to offload the computation to the cloud. As a consequence, for the effectiveness of the computation offloading, offloading decisions must consider all the different costs incurred by the offloading. In addition, the context of the user (e.g., the battery level of his mobile device and his mobility) changes permanently and has significant effect on his preference for the offloading decisions.

Despite the significant amount of work, which has been performed on computation offloading, most of the existing offloading frameworks focus on what components of the application to offload, how to offload the components and where to offload the intensive component of the application (Shiraz et al., 2015). Many of those works focus on designing frameworks to address either one or multiple offloading criteria (Chun et al., 2011; Cuervo et al., 2010; Giurgiu et al., 2009; Kemp et al., 2010; Kosta et al., 2012; Kumar & Lu, 2010; Miettinen & Nurminen, 2010; Wolski et al., 2008; Xia et al., 2014; X. Zhang et al., 2011). Among the popular criteria are: minimizing energy consumption (Cuervo et al., 2010; Kumar & Lu, 2010; Miettinen & Nurminen, 2010), reduction in execution time (Giurgiu et al., 2009; Wolski et al., 2008). In addition to those, there are also combinations of those objectives: minimizing energy and execution time (Chun et al., 2011; Kosta et al., 2012; Xia et al., 2014), and minimizing execution time, cost, and network latency(X. Zhang et al., 2011). Although a few approaches proposed a context-aware offloading scheme for mobile cloud computing information(Ghasemi-Falavarjani et al., 2015; Lin, Lin, Hsu, & King, 2013; B. Zhou, A. V. Dastjerdi, et al., 2015; B. Zhou, Vahid Dastjerdi, Calheiros, Srirama, & Buyya, 2015), they did not consider all the costs

incurred by the offloading, and the user context and his preference for making the offloading decision.

Our work goes beyond the state-of-the-art by defining an offloading decision making model that minimizes the offloading costs, subject to multiple constraints including application, smartphone, network, and cloud constraints while considering the user context and user preferences. However, no-specific linear equation can describe the relationship between all those factors that influence the offloading decision. Therefore, we need to develop an offloading decision making strategy for computation offloading. This strategy could help in making right decisions as to whether to perform computation offloading or not, based on different cost of the offloading and user context and user preference.

Although it can be assumed that users can consider all various combinations of those factors and make a good decision, the frequency of those decisions will be cumbersome for the user. Therefore, user decisions need to be supported for making optimal offloading decision in the long run.

Therefore, our research objective can be expressed through the following three research questions:

How can we combine time cost, energy consumption cost, communication cost, and computation cost, and user context for the offloading decision?

How can the offloading decision be supported through an optimization method that is based on the user preferences?

What is the performance of such an optimization method?

The main contribution of this paper are as follows: First, an offloading decision making framework is proposed, which aims at minimizing different aspects

of the offloading cost while considering user context (e.g., his mobility and the energy remained in his smartphone). Second, the user context and different costs (i.e., monetary cost, time cost, energy cost, communication cost, and computation cost) are integrated through the use of user preferences. Third, the proposed framework employs a novel approach for making offloading decision based on Deep Learning Neural Networks [DNNs] to support the user in making optimal offloading decisions.

Our simulation results indicate that implementing a learning mechanism such as DNN to capture user preference for the computation offloading could help to optimize the offloading decision taking into consideration all the factors that affect the offloading. The user needs to give little feedback to the neural network to make optimal offloading decision based on different cost and context factors.

The remainder of this paper is organized as follows: related work is described in Section 4.2. In Section 4.3, we present the proposed model for computation offloading decision making and the offloading decision making algorithm. In Section 4.4, we evaluate the performance of the model through applying it to the neural networks. The results are also presented in this section, while Section 4.5 concludes the paper with a brief evaluation.

4.2 Related Works

The concept of offloading the computation and data to the cloud is used to address the limitation of mobile devices by using the resources provided by the cloud rather than the mobile device itself to run the mobile application (Fernando et al., 2013). There is a significant amount of computation offloading frameworks

which were designed to address one objective criteria of off-loading (Chun et al., 2011; Cuervo et al., 2010; Giurgiu et al., 2009; Kemp et al., 2010; Kosta et al., 2012; Kumar & Lu, 2010; Miettinen & Nurminen, 2010; Wolski et al., 2008; Xia et al., 2014; X. Zhang et al., 2011). Among the popular objectives are: minimizing energy consumption (Cuervo et al., 2010; Kumar & Lu, 2010; Miettinen & Nurminen, 2010) , reduction in execution time [(Giurgiu et al., 2009; Wolski et al., 2008). For example, a simple analytical model helps deciding whether to offload the computation or not by comparing energy usage in the cloud and on the mobile device(Kumar & Lu, 2010). They present energy models that trade-off the energy consumption on the mobile device versus the energy needed for the computation offloading to the cloud. In addition to those, there are also combinations of those objectives: minimizing energy and execution time (Chun et al., 2011; Kosta et al., 2012; Xia et al., 2014), and minimizing execution time, cost, and network latency (X. Zhang et al., 2011).

In two previous offloading studies, the bandwidth was the only network parameters considered for the offloading decision (Kumar & Lu, 2010; Wolski et al., 2008). The authors also assumed power consumption that is required for sending and receiving data is same, but another study by Feeney and Nilsson showed that wireless network interfaces exhibit a complex range of consumption behaviors (Feeney & Nilsson, 2001). Hence, another factors such as packet size, the number of broadcasts, and point-to-point traffic need to be considered when designing energy-aware offloading algorithms. (Wen, Zhang, & Luo, 2012) considered the wireless channels features and showed that execution policies depend on the input data size

and completion deadline for the application, as well as the wireless transmission model.

Although a few approaches proposed a context-aware offloading scheme for mobile cloud computing information (Ghasemi-Falavarjani et al., 2015; Lin et al., 2013; B. Zhou, A. V. Dastjerdi, et al., 2015; B. Zhou, A. Vahid Dastjerdi, et al., 2015), they did not consider all of the cost factors which incurred by the offloading , the context of the user, and his preference. For example, (Lin et al., 2013) proposed a context-aware decision algorithm, called CADA , to optimize the performance of the mobile device with various optimization criteria, including short response time and low energy consumption. The proposed system profiles the user location and time of the day when tasks are offloaded for remote execution. The algorithm takes the decision whether to offload tasks or not based on historical log records such that the task is offloaded if its energy consumption when it was offloaded in the past is lower than when it was executed locally for the same time of day and the same geographic location. Such approach that depends on historical records might not be suitable for present conditions and could lead to inaccurate offloading decisions.

(B. Zhou, A. V. Dastjerdi, et al., 2015; B. Zhou, A. Vahid Dastjerdi, et al., 2015) propose a context-aware offloading decision algorithm that aims to provide code offloading decisions at runtime on selecting wireless medium and which potential cloud resource (nearby mobile cloud, cloudlet, or public cloud VMs) as the offloading location based on the device context. They present cost estimation models for each of those offloading location, and use the estimation results and device context to provide offloading policies of where, when and how to offload for

the mobile application. The cost estimation model includes the energy consumption and task execution time but didn't include the monetary cost.

(Ghasemi-Falavarjani et al., 2015) investigated the resource allocation problem in mobile cloud. They developed a context-aware offloading middleware for mobile cloud (OMMC) to collect contextual information of mobile devices, subtasks, and environmental variables, and also to manage the offloading process. However, regarding resource allocation, it only optimize the energy consumption and execution time of offloading while considering some constraints such as user's acceptable deadline, service providers' residual energy and budget constrain.

(Magurawalage et al., 2014) proposed system architecture for mobile cloud computing (MCC) that includes a cloudlet layer located between mobile devices and their cloud infrastructure or clones. This middle layer is called a cloudlet layer as it composed of cloudlets. Cloudlets are deployed next to IEEE 802.11 access points and serve as a localized service point closed to mobile devices to improve the mobile cloud services performance. They proposed an offloading algorithm on top of this architecture with the purpose of deciding whether to offload to a clone or a cloudlet. The decision-making considers the energy consumption for task execution and the network status while satisfying the response time constraints of the certain task.

(Kovachev et al., 2012) use integer liner programming to address adaptive computation offloading as an optimization problem. Their approach considers energy usage, available memory and CPU as the criteria for offloading. The algorithm adapts dynamic approach to make the offloading decision by solving a new optimization problem each time parameters such as the available bandwidth and

memory updated their values in the model. Another work by (Wu, 2013) takes into consideration network unavailability to make the offloading decision. The model uses an application partitioning algorithm, and an offloading decision module intelligently decides on whether to offload by considering the network availability for remote execution. (Ou, Yang, & Hu, 2007) proposed CRoSS algorithm to selects the best host for offloading according to the link cost. The cost of the link includes both the link failure rate and the bidirectional transmission rate.

Compared to the related works, our study considers different costs (i.e., time cost, energy cost, and monetary cost), and the importance of the user context such as his mobility and amount of remained energy as well as the user preference to make the offloading decision.

4.3 proposed Model for computation offloading decision

4.3.1 Model Architecture

As mentioned, our work concentrates on the development of an offloading decision strategy for the computation offloading. This strategy could help in making right decisions as to whether to perform computation offloading or not based on different costs of the offloading while considering user context and preference. An illustration of the architecture of the proposed model is presented in Fig.12.

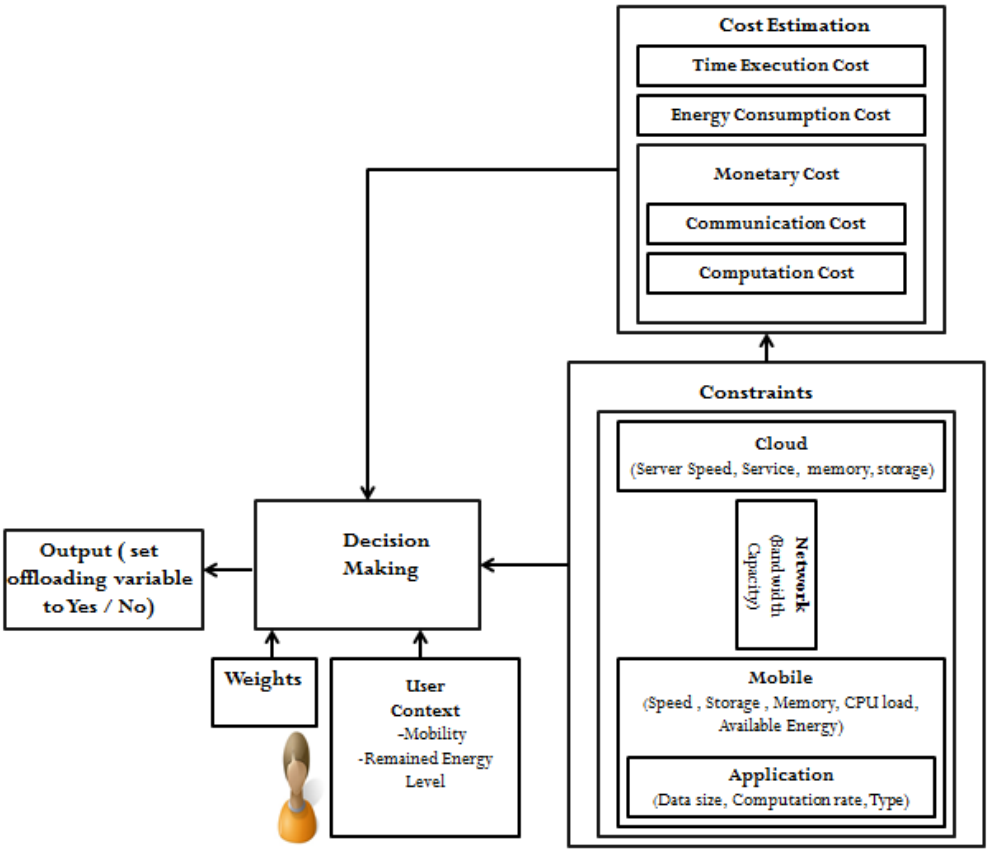


Figure 12. Model architecture

, cost estimation model, user context, and decision making. We will describe each of them in the following subsections.

1) Constraints: constraints are a critical part for the offloading decision as they represent the available resources for running the application including:

a) Cloud constraints: Such as the speed of cloud server S, cloud server storage H, cloud server memory Z, and the cost charged by the cloud service provider for the computation. All the symbols are listed in Table 4.

b) Network constraints: When making the offloading decision it is necessary to consider the state of the available network bandwidth between the smartphone and the cloud. We consider two modes of network connection between

the smartphone and the cloud. Those are WiFi network and cellular network (3G or 4G). The network bandwidth capacity varies according to the type of the used network. It also varies according the current usage of the link between the smartphone and the cloud.

c) Mobile constraints: Including mobile speed M, CPU load, memory A, and storage K. In case of the CPU, the mobile CPU can be either idle, or have utilization from 0-100%. Therefore, this constraint should be considered when making the offloading decision.

d) Application Constraints: such as data size D, Computation rate C, and application type. For example, some applications (e.g., chess game) have small size of data but the amount of computation is extremely large. Whereas other applications (e.g., image retrieval) have large size of data but the amount of computation is very small (Kumar & Lu, 2010). Therefore, the application type has an influence on the offloading and need to be considered in the offloading decision.

Table 4. Notation

Variable	Description	Unit
C	Computation size of an app	I(instructions)
S	Cloud server speed	I /s
M	Mobile speed	I/s
H	Cloud storage	Byte
Z	Cloud memory	Byte
A	Mobile memeory	Byte
K	Mobile storage	Byte
$\boldsymbol{B_{send}}$	Network Bandwidth for sending the data to the cloud	B/s
$\boldsymbol{B_{receive}}$	Network bandwidth for receiving the data from the cloud	B/s
$\boldsymbol{D_{send}}$	Data size need to be transmitted to the cloud	B

$D_{receive}$	Data size need to be transmitted to the cloud	B
E_{local}	Local energy consumption cost	J
E_{cloud}	Offloading energy consumption cost	J
T_{local}	Time cost for local execution on mobile	s
T_{cloud}	Time cost for executing the application on the cloud	s
T_{send}	Time for transmitting data to the cloud	s
T_{idle}	Waiting time for the execution on the cloud	s
$T_{receive}$	Time for receiving the results from the cloud	s
P_{local}	Power consumption for local execution	W=J/s
P_{send}	Power consumed for transmitting data	W=J/s
P_{idle}	Power Consumed for waiting	W=J/s
$P_{receive}$	Power consumed for receiving the result	W=J/s
C_{money}	Monetary cost	$
C_{comm}	Communication Cost	$
C_{comp}	Computation cost	$
CPU_{cloud}	Cloud cpu required to run the app	cycles
$P_{cprocess}$	Price of the cloud CPU cycle	$/cycle
P_H	Price of the cloud storage	$/B

2) Cost Estimation: Before the smartphone runs the application, the cost estimation model predicts the cost for running the application locally on smartphone as well as the offloading cost for the executing the application on the cloud. The cost estimation component used calculates the different costs based on the constraint

values passed from the constraint component and passes them to the decision making component.

The cost estimation consists of three subcomponents:

(i) **Time cost:** measures the time required to execute the application locally on the smartphone and the time required to execute the application on the cloud. The equations used in estimation the time cost are based on (Kumar & Lu, 2010; Xia et al., 2014) as follows:

Local time cost (T_{local}): This is the time required to run the application locally on the smartphone. It is estimated value depends on the amount of the computation C, and the smartphone speed M

$$T_{local=C/M} \tag{4-1}$$

Offloading time cost (T_{cloud}): This is the time required for the running the application on the cloud. It comprises of time required for transmitting the required data for the application, waiting the result, and receiving execution from the cloud. The calculation of the time cost depends on the size of the data to be sent and received, the bandwidth capacity for sending and receiving the data, the computation rate, and the server speed of the cloud. The equations used to calculate the offloading cost are based on (Kumar & Lu, 2010)and (Xia et al., 2014) as follows:

$$T_{cloud} = T_{send} + T_{idle} + T_{receive} \tag{4-2}$$

$$T_{send} = D_{send}/B_{send} \tag{4-3}$$

$$T_{idle} = C/S \tag{4-4}$$

$$T_{receive} = D_{receive}/B_{receive} \quad (4\text{-}5)$$

(ii) Energy consumption cost : To support the offloading decision, the energy consumption cost for local and offloading execution of the application need to be estimated before deciding whether to offload or not. The equations used in estimation the energy consumption cost are similar to that of (Kumar & Lu, 2010)and (Xia et al., 2014) as follows:

Local energy consumption cost ($\boldsymbol{E_{local}}$): This is the energy consumed by executing the computation locally on smartphone can be calculated based on the power consumed ($\boldsymbol{P_{local}}$) and the average time required to run the application on the smartphone ($\boldsymbol{T_{local}}$). i.e. :

$$E_{local} = P_{local} * T_{local} = P_{local} * \text{ C/M} \quad (4\text{-}6)$$

Offloading energy consumption cost ($\boldsymbol{E_{cloud}}$): This cost is the energy consumed by mobile device for executing the application on the cloud. This includes the energy consumed for transmitting the required data of the application to the cloud, waiting for the cloud to complete the execution, and receiving the result from the cloud. Therefore:

$$E_{cloud} = E_{send} + E_{idle} + E_{receive} \quad (4\text{-}7)$$

$$E_{send} = P_{send} * T_{send} \quad (4\text{-}8)$$

$$E_{idle} = P_{idle} * T_{idle} \quad (4\text{-}9)$$

$$E_{receive} = P_{receive} * T_{receive} \quad (4\text{-}10)$$

(iii) Monetary Cost (C_{money} y): As mentioned, offloading the application to the cloud requires cost for the smartphone user based on the consumed resources for communicating and running the application on the cloud. It comprises communication cost and computation cost. In this paper, we only consider the monetary cost for the offloading case since local execution of the application uses the mobile resources without need to connect to the cloud and hence it doesn't have any monetary cost.

Communication cost (C_{comm}): This involves cost of wireless/ mobile networks and the cloud. User can offload his application to the MCC by either accessing WiFi or using 3G/4G mobile network. However, he can reduce the cost by using WiFi because wireless networks such as 3G links will charge him more cost. The communication cost is based on the amount and price of the data traffic for the communication required to execute the application on the cloud. This cost is determined by the service agreement between the user and the network provider.

Computation cost (C_{comp}): This is the usage cost of the cloud resource. Different cloud service providers charge different rates for the same service. Usually, cloud service provider measures the computation cost based on the amount of CPU cycles, storage, and communication traffic (in and out) of a cloud (X. Zhang et al., 2011). In this work, we assume that the computation cost is calculated based on the amount of CPU cycles (CPU_{cloud}) and storage (H) and their corresponding prices.

$$C_{comp} = CPU_{cloud} * P_{cprocess} + H * P_H \quad (4\text{-}11)$$

The total monetary cost for offloading the application can be estimated as:

$$C_{money} = C_{comm} + C_{comp} \quad (4\text{-}12)$$

In some cases, the user can offload his application to the cloud using free WiFi, in such cases the cost is considered to be zero.

3) User context: Given the cost estimation for both local execution and offloading, the user context information such as his mobility and the remained energy level should be considered when making the offloading decision. In this work the user takes into account his context (e.g., mobility and energy level) as they are considered to have a significant effect on the offloading decision. For example, when the remained energy in the smartphone user is low and have no access to charge his mobile, in this case he would prefer to offload the heavy computation tasks to be executed on the cloud to save the battery energy. Also, in case of the user mobility, a user may lose connectivity to the internet because of changing his location but might reconnect again later after short time or might not reconnect to the internet for a long time. Therefore, the availability of the connectivity for the mobile user varies according to the current location of the user. Also, the availability of the network connectivity and its data rate for the mobile user may change during the remote execution time for one application (Magurawalage et al., 2014). In

addition, disconnection during the application execution can negatively affect the user experience.

4) Decision making: This is the core component of our model as it decides whether to offload the application execution to the cloud or not. As mentioned before, our work focuses on the development of an offloading decision strategy for the computation offloading to help in making optimal offloading decision based on different costs of the offloading while considering user context and preference. Therefore, decision making is not a fully automatic but it mainly depends on the capability of the user to make a complex optimal offloading decision based on different cost and context factors. For the purpose of the model examination, we assume that, the decision making by the user can be based on presenting the cost values which estimated by the cost estimation model, to the smartphone user. The offloading decision aims at minimize the execution cost while considering the user context and preference. The user preferences will help to weight the different cost factors with respect to the user's context. For example, when the user considers the energy consumption is more important due to the limited energy level in his battery, he may assign higher weight to the energy consumption cost. The workflow of the offloading decision making is presented in the next section.

4.3.2 Offloading decision making Algorithm

Given the cost estimation method in the previous section, the workflow of offloading decision algorithm is presented in Figure 13. It needs steps to make the offloading decision:

Step1: It predicts the execution time required for running the application locally on the smartphone T_{local} .

Step2: It predicts the execution time required for running the application on the cloud T_{cloud}.

Step3: It estimates the energy consumption of running the application locally on the mobile, E_{local}.

Step4: It estimates the energy consumption of running the application on the cloud E_{cloud}.

Step5: It calculates the monetary cost for running the offloading the application on the cloud C_{money}

Step6: It compares the sets of time and energy costs of local and offloading (T_{local} and T_{cloud}, E_{local} and E_{cloud}) as well as considering the monetary cost C_{money}.

Step7: Given the user context, the user may have to assign different weights to the cost factors and make the offloading decision based on his preference.

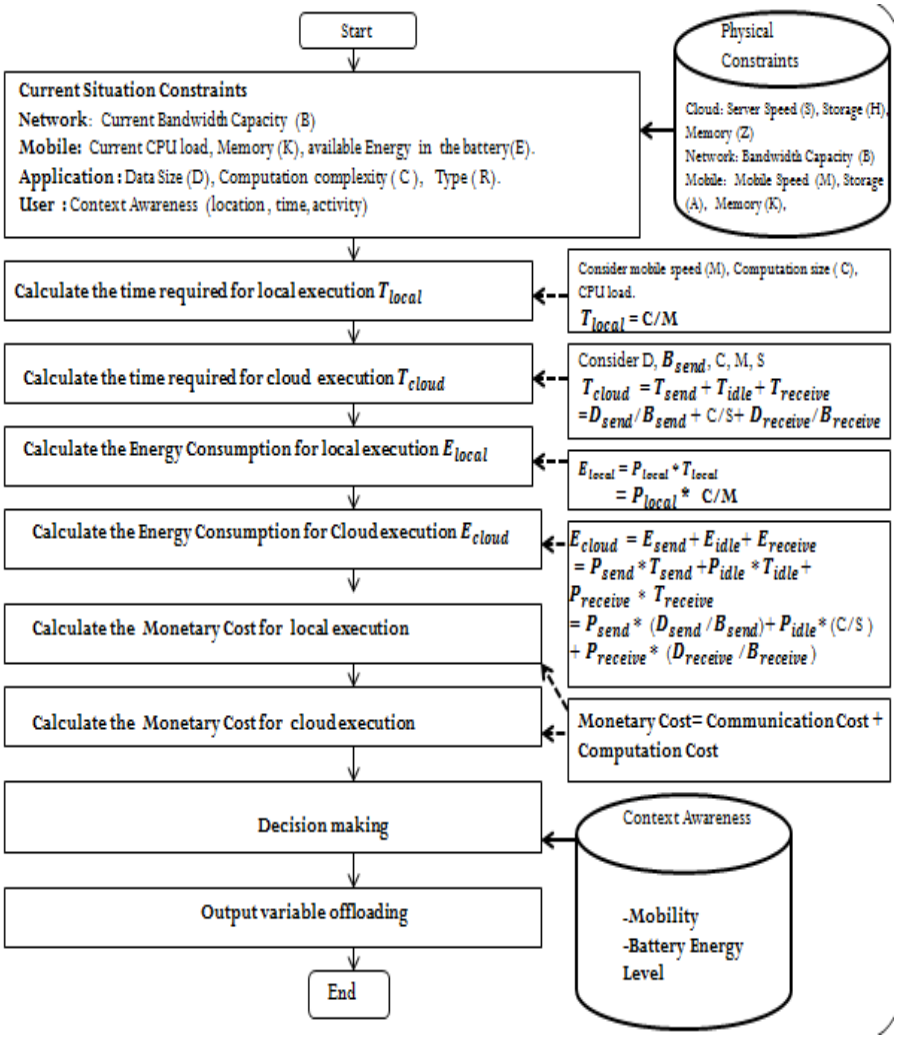


Figure 13. Workflow of the offloading decision making

4.4 Implementation of Deep Neural Network (DNN) to support the user in making offloading decision

4.4.1 Experiment

In this section, we apply the neural networks for supporting smartphone users in making offloading decision through a use case.

4.4.1.1 Data

Considering the different cost and user context variables that effect the offloading decision used in our proposed model (local energy cost, offloading energy cost, local time cost, offloading time cost, local monetary cost, offloading monetary cost, user's mobility, and remaining energy in the smartphone battery), we assumed that, a user has provided with the values of the those factors and we asked the user how he will response if he becomes a part of the offloading decision making. The key of optimal offloading decision is the accuracy of user decision. We considered the offloading decision as a binary classification problem where the input data are the values of the 8 variables and the output is either to offload or not to offload. Therefore, there are two classes of the output (class 0 not offload, and class 1 to offload). We used a frequency of 2000 examples of offloading decisions made by the user as our training data set.

4.4.1.2 Network Structure

To build a neural network to perform some task, one must decide about the choice of network structure such as how many hidden layers to be used, how many neurons in each hidden layer are needed, and how the neurons are to be connected. That is, wrong choice of network structure can lead to poor performance. If you choose a network that is too small, then them model will be incapable to represent the desired function. On the other hand, choosing a big network will make the model able to memorize all the examples of training data but it may not be able to generalize well to the unseen data. The problem of choosing the right number of

hidden neurons in advance is still not well-understood. Therefore, it is more common to see hill-climbing searches that selectively modify an existing network structure. There are two ways to do this: start with a big network and make it smaller, or start with a small one and make it bigger (Russell, S., et al., 1995).

In this work a feed-forward network is used that has more than one hidden layer between the input and the output. In a layered-forward network, each neuron is linked only to neurons in the next layer; there are no links between neuron in the same layer, no links backward to previous layer, and no links that skip a layer.

For our experiments, two NNs structures were used with two hidden layers and but with different numbers of neurons in the hidden layers. In the first experiment we used a neural network with tow hidden layers each of 64 size, referred as 8-64-64-1 (Figure 14).

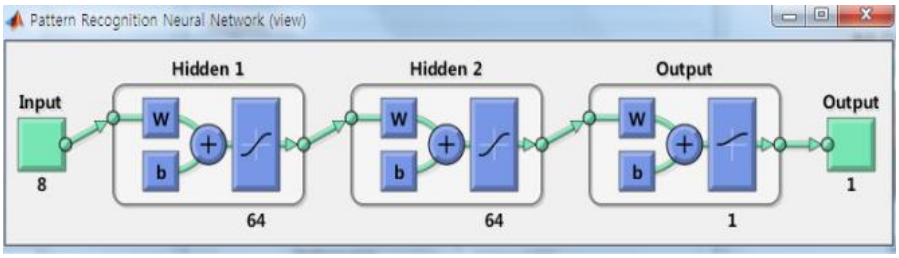


Figure 14. A 8-64-64-1 Pattern Recognition Neural Network (view)

And in the second experiments we used a neural network with two hidden layers one of 64 size and the other one is of 32 size(Figure 15)

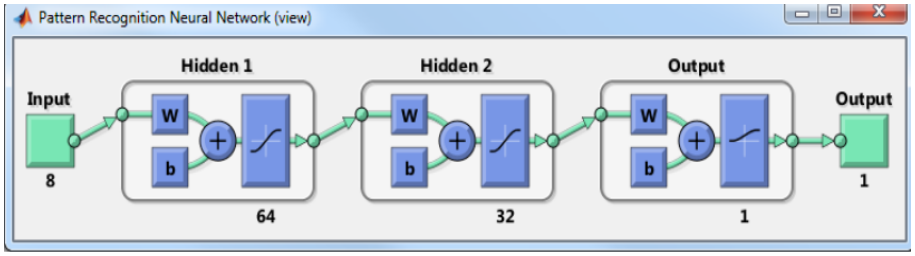


Figure 15. A 8-64-32-1 Pattern Recognition Neural Network (view)

Each hidden layer uses the *tansig* neural transfer function to calculate the layer's output from the layer before it and send it to the next layer.

The *tansig* neural function generate output between -1 and 1 as the neuron input goes from negative to positive infinity.

$$a_i = tansig\ (in_i) = \frac{2}{(1+e^{(-2*in_i)})-1} \qquad (4\text{-}13)$$

As our networks are a binary classification networks (only a single output neuron is used), and the common choice for the transfer function of the output of the binary classification is the logistic activation function , which always generates results in the range 0 to 1 (Edward R. Jones, 2004). Therefore, at the output layer we use *logsig* neural function. The *logsig* neural function generate output between 0 and 1 as the neuron's net input goes from negative to positive infinity.

$$a_i = logsig\ (in_i) = \frac{1}{(1+e^{(-in_i)})} \qquad (4\text{-}14)$$

4.4.1.3 Training with the Back Propagation Algorithm

Over the last years, many hundreds types of Neural Network have been proposed. There are many different names for the neural networks such as Artificial Neural Networks (ANNs), Connectionism or Connectionist Models, Multi-layer perceptron's (MLPs) and Parallel Distributed Processing (PDP). Despite all the different types and different terms, there are small group of "classic" networks which are widely used and on which many others are based. These are: Back Propagation, Hopfield Networks, Competitive Networks and networks using Spiky Neurons. In this work we will use the back propagation network.

A Back Propagation network learns by example. By giving the algorithm examples of what we want the network to do, it changes the network's weights so that, when the training is finished, the network will give the required output for a particular input.

The goal of backpropagation is to compute the partial derivatives of the cost function that measures the discrepancy between the target outputs and the actual outputs produced for each training case (Rumelhart, David E, et al., 1988).

The training of the network by backpropagation consists of three stages:

- The feedforward of the input training pattern.
- The calculation and backpropagation of the associated error
- The adjustment of the weights so that the output from the network approximates the target output for all training inputs.

First apply the inputs to the network and calculate the output. As the initial weights and biases are random numbers (usually from the range [-0.5,0.5], this output will

not match the target output. In the backpropagation model, the error between the calculated output and target output is minimized through iterative updates of weights for all training patterns (epochs). The process of adjusting the network weights until the calculated output approximates the target output is called the leaning method. The generic neural network learning method is depicted in Figure 16.

```
function NEURAL-NETWORK-LEARNING(examples) returns network

  network ← a network with randomly assigned weights
  repeat
     for each e in examples do
        O ← NEURAL-NETWORK-OUTPUT(network, e)
        T ← the observed output values from e
        update the weights in network based on e, O, and T
     end
  until all examples correctly predicted or stopping criterion is reached
  return network
```

Figure 16. The generic neural network learning method (Source: (Russell, S., et al., 1995).

In practice, this approach enables the network to have a good performance but slow convergence to the final outcome. Therefore, in order to accelerate the BP algorithm and instead of minimizing the squares of the differences between the calculated and target , we used the cross entropy error function to be minimized:

$$E^c = -\sum_{i=1}^{N}\{t_i \ln\ (\hat{t}_i) + (1 - t_i)\ln\ (1 - \hat{t}_i)\} \qquad (4\text{-}15)$$

Where N is the number of training patterns (examples), t_i is the target value for the *ith* case (either 1 or 0 , and $\hat{t}$ is the network calculated output for the *ith* case).

In multilayer networks, there are many weights connection connecting each input and each output, and each of these weights contributes to more than one output. To reduce the error between the target output and the calculated output, the backpropagation makes small changes in the weights. The back-propagation algorithm for updating weights in multi-layer network is shown in Figure 17. It can be summarized as follows :

- The calculation and backpropagation , A value, of the associated error.
- Starting with output layer, repeat the following for each layer in the network, until the earliest hidden layer is reached:

  -Propagate the A values back to the previous layer

  - Update the weights between the two layers.

```
function BACK-PROP-UPDATE(network, examples, a) returns a network with modified weights
  inputs: network, a multilayer network
          examples, a set of input/output pairs
          α, the learning rate

  repeat
    for each e in examples do
      /* Compute the outputfor this example *I
        O ← RUN-NETWORK(network, I^e)
      /* Compute the error and Δ for units in the output layer */
        Err^e ← T^e - O
      /* Update the weights leading to the output layer */
        W_{j,i} ← W_{j,i} + o x a_j x Err_i^e x g'(in_i)
      for each subsequent layer in network do
        /* Compute the error at each node */
          Δ_j ← g'(in_j) Σ_i W_{j,i} Δ_i
        /* Update the weights leading into the layer *I
          W_{k,j} ← W_{k,j} + a x I_k x Ay
      end
    end
  until network has converged
  return network
```

Figure 17. The back-propagation algorithm for updating weights in multi-layer network (Russell, S., et al., 1995).

4.4.2 Evaluation and performance

In this section, the evaluation and the performance of the two NNs are discussed. There are many ways of measuring the network performance. Confusion matrix and cross-Entropy error are the most popular metrics.

4.4.2.1 Confusion Matrix

The confusion matrix focus is on the predictive capability of a model [1].The confusion matrix is represented by a matrix which each row represents the instances in a output(actual) class, while each column represents in an target (predicted) class. One of the advantages of using this performance evaluation tool is that the data mining analyzer can easily see if the model is confusing two classes (i.e. commonly mislabeling one as another).

The matrix also shows a detailed breakdown of correct and incorrect classifications for each class. The accuracy of the classifier is presented as the percentage of correctly classified patterns in a given class divided by the total number of patterns in that class. The overall (average) accuracy of the classifier is also evaluated by using the confusion matrix.

Based on the training data of the 8-64-64-1 NN, the result of the confusion matrix is presented in Figure 18. The matrix shows that, of the 1443 patterns that are class 0 (Not to offload), 1358 were correctly classified as " Not to offload " while 85 were incorrectly classified as " To offload" (achieving accuracy of 94.1 %). On the other hand, of the 556 patterns that are class 1 (To offload), 496 are

[1] http://aimotion.blogspot.kr/2010/08/tools-for-machine-learning-performance.html

correctly classified as “To offload” while 60 were incorrectly classified as “ Not to offload” (an accuracy of 89.2 %). The overall accuracy of this network for making the offloading decision given this data set of 1999 patterns is evaluated achieving 92.7 %.

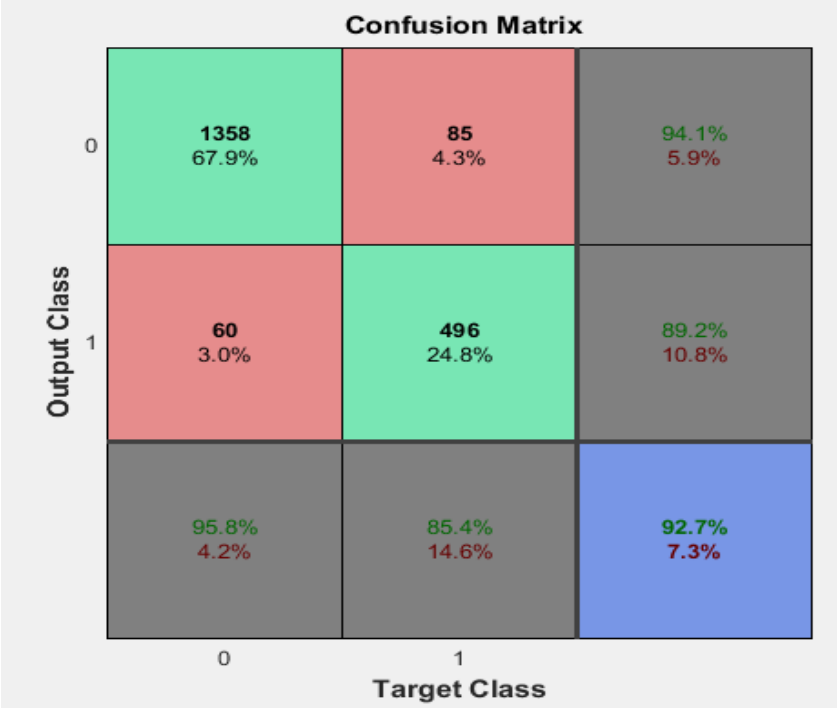


Figure 18. Confusion Matrix for the 8-64-64-1 NN.

In neural networks, each time a network is trained, can result in a different output due to different division of dataset into training, validation, and test sets. and also due to different initial weight and bias values. As a result, different neural networks trained on the same problem can give different output for the same input. To ensure the accuracy of using the neural network in making the offloading decision, we train another neural network with different architecture (8-64-32-1) using the same dataset. The results of the confusion matrix for the 8-64-32-1 NN are presented in Figure 19. The matrix shows that, of the 1470 patterns that are class

0 (Not to offload), 1362 were correctly classified as " Not to offload " while 108 were incorrectly classified as " To offload" (achieving accuracy of 92.7 %). On the other hand, of the 529 patterns that are class 1 (To offload), 473 are correctly classified as "To offload" while 56 were incorrectly classified as " Not to offload" (an accuracy of 89.4 %). The overall accuracy of this network for making the offloading decision given this data set is evaluated achieving 91.8 %.

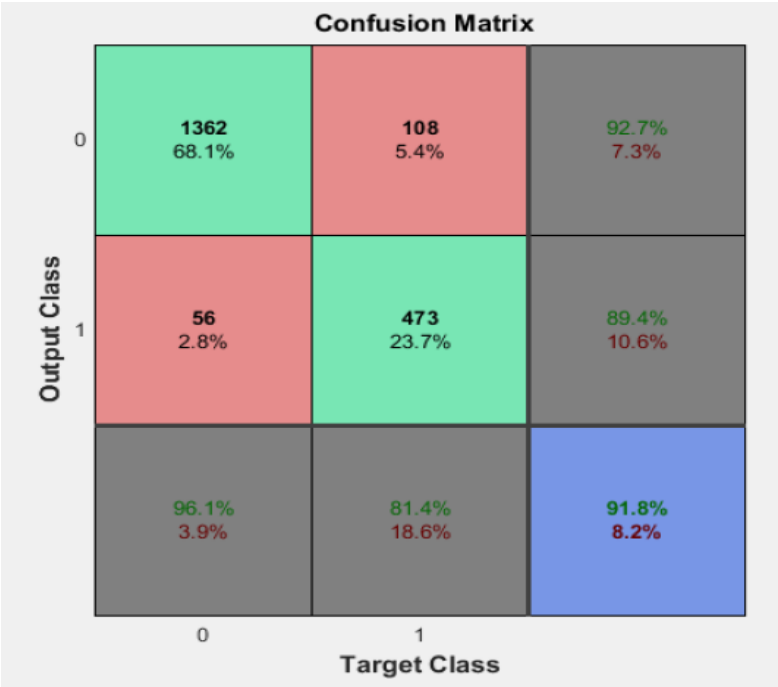


Figure 19. Confusion Matrix for the 8-64-32-1 NN

4.4.2.2 Cross Entropy Error

To train a neural network we need some measure of error between computed outputs and the desired target outputs of the training data. The most common measure of error is called mean squared error. However, there are some

research results that suggest using a different measure, called cross entropy error, is sometimes preferable to using mean squared error[2]. The cross-Entropy error is calculated using Eq (4-15).

The cross-entropy plot for the 8-64-64-1 NN in Figure 20 shows how the training proceeds on the training, validation and evaluation sets .Training cross-Entropy shows a good training and the validation and test curves are very similar.

The figure also shows that, as we can see in the performance plot, with the epochs , the cross-entropy error of the NN has decreased with the best validation performance of 0.2387 at epoch 18.

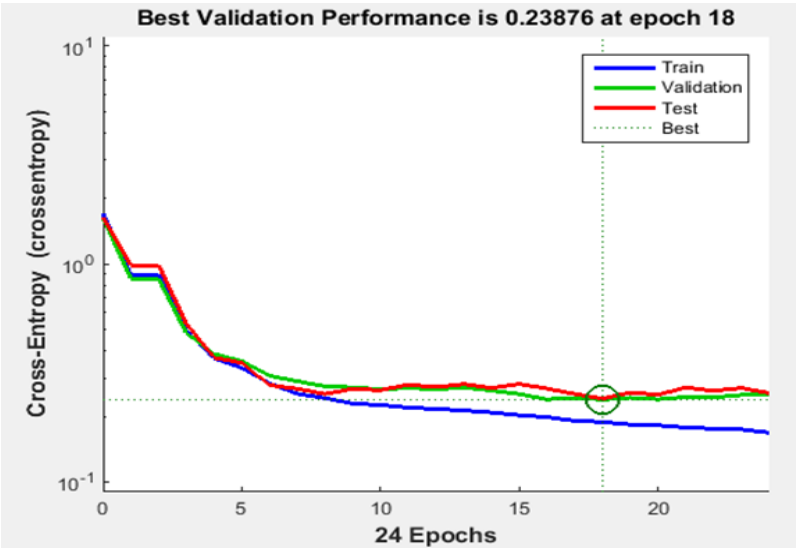


Figure 20. performance plot for 8-64-64-1 NN

For the 8-64-32-1 NN the performance plot in Figure 21 doesn't indicate any problem with the training. It shows that all the training, validation and test curves are very similar. The performance plot, also shows that, the cross-entropy

[2] https://visualstudiomagazine.com/articles/2014/04/01/neural-network-cross-entropy-error.aspx

error of the NN has decreased with the best validation performance of 0.2339 at epoch 13.

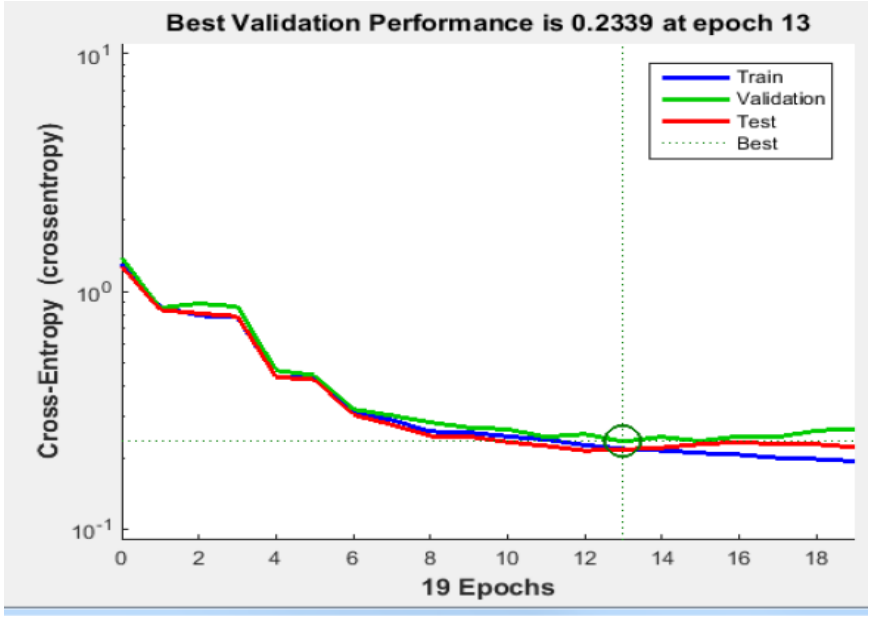


Figure 21. Performance plot for 8-64-32-1 NN

4.4.2.3 Testing the Network

As the network is trained, and it performance evaluated shows the high accuracy of the network, we test the network with new input data to evaluate its prediction for the offloading decision get. By applying a new input to the network with the same format of the input variables, it output was accurate. Figure 22 predicts an example of using the network with new data which isn't seen by the network.

```
K>> tt = [0.1000    0.3100  0.1300  0.5700  0   0.7900  0.4700  0.5400]';
K>> net(tt)

ans =

   4.0956e-04
```

Figure 22. Testing the trained network

As it shown the network output is very small (0.00040956). This number is closer to 0 than 1, so we can approximate it as label 0. In which case it means not to offload. Comparing the offloading decision made by the neural network with the offloading decision examples made by the user shows the accurate performance of the neural network. That is, the neural network could learn and make a decision which matches the user preferences.

4.5 Conclusion

In this chapter, we proposed an offloading decision making algorithm which aims at minimizing different aspects of the offloading cost while considering user context (e.g., his mobility and the energy remained in his smartphone). We integrated the user context and different costs (i.e., monetary cost, time cost, energy cost, communication cost, and computation cost) through the use of user preferences. We evaluated the potential of neural networks for supporting the user in making offloading decision through a use case. The Performances of the networks indicate that little feedback from the user can be used to train the network and achieve optimal offloading decisions at a high accuracy.

This study proofs that, based on the human's cognition, the neural network can be trained, which then will be able to mimic the decision making criterion of the human user. Using the neural network can support the smartphone user in making a complex offloading decision that will minimize the offloading cost and match the user expectations.

Chapter 5. Conclusion and Implications

5.1 Summary

This dissertation discussed two main aspects that deal with the overcoming the resource limitations of smartphone. First a utility-based energy consumption optimization model was proposed, which considers the user's preferences with respects to energy consumption aspect when using smartphone applications. The user's utility function for the energy allocation to the two types of smartphone application uses (i.e on device application use and cloud-based application use) is defined by the Cobb-Douglass function . By applying this model to a real case study, the results show that, by taking into account user preferences, an optimal (i.e., utility maximizing) allocation of energy to different types of application uses is possible for a user. The allocation by the user determines not only the ratio between the two types of application uses but also the quantity with which the energy in the battery should be consumed. The user only needs to specify the perceived cost of energy for each type of application. The study also examines the effect of the user feedback about his preferences for allocating the energy on his utility. The result shows that, the user utility is improved as the user provides feedback and the utility improvement increases with the increased frequency of the user feedback. On the other hand, the results show that if the user has no intervention about energy allocation his utility decreases linearly with respect to energy drain.

In the second assay, we proposed an offloading decision making algorithm which aims at minimizing different aspects of the offloading cost while considering

user context (e.g., his mobility and the energy remained in his smartphone). We integrated the user context and different costs (i.e., monetary cost, time cost, energy cost, communication cost, and computation cost) through the use of user preferences. We evaluated the potential of using neural networks for supporting the user in making a complex offloading decision. Using data about offloading decisions examples made by user based on those different cost factors and user context, two neural networks with different architectures were trained and evaluated . After training, the performance of the two neural networks indicates that little feedback from the user can be used to train the network and achieve optimal offloading decisions at a high accuracy.

5.2 Discussion and Implications

The results of the first essay show that the energy consumption can be adjusted to maximize the user utility. The user preferences are reflected through the past patterns of energy consumption by the two type of application uses (i.e on-device application uses and cloud-based application uses), the perceived costs of energy allocation for the different types of applications. The past patterns of energy consumption is of interest because it can give indications about the user preferences. For example, a particular user may always use his smartphone for social networking applications, if he is moving and have no chance to charge his battery soon, then he should pay intention to the energy allocation between the applications. Therefore, he can maximize his utility by consuming less energy for other kinds of applications and use it for the applications of his preferences.

These results provide some additional interesting understanding of smartphone usage behavior, and it will contribute to the line of studies assessing the effect of usage behavior on energy consumption optimization of smartphones. It can be used by smartphone users to obtain the maximum utility from the remaining energy in their smartphones battery. And also it can be used by the smartphone manufactures and developers to enhance the smartphone's platform. They should recognize that there is a need to implement a learning mechanism to monitor user behavior and learn his preference and provide the user with a feedback that could help him to adjust his usage behavior and take proactive measures to allocate the energy consumption between the different types of the applications uses. In addition, considering the user's sensitivity to energy consumption by some applications, the smartphone applications developers should allocate the energy-intensive computation to the cloud instead of smartphone, thus helping to extend the battery lifetime.

We argue that there exists a potential in optimizing the energy consumption of smartphone battery by better understanding user's preferences in using the applications. By analyzing user's preference, we can assess the extent to which the user can allocate the energy consumption to the applications of his preference, and explore how the user utility can be maximized from the remaining energy in his battery.

Concerning the computation offloading decision, making an optimal offloading decision which considers all the different cost incurred by the offloading process as well as the user context factors is very complex task. However, our results of the second study show that the implementation of learning technique, such

as the neural network, will help in making such decision that can minimize the cost and match the user expectations form the offloading process. Such results provide new insights for the smartphone manufactures and developers. They should recognize that they should give the users control over specific tasks such as whether to execute the application locally on device or offload it to the cloud in order improve the usage experience for the smartphone user.

5.3 Limitations

Finally, it is important to note about the limitation of this study. Our findings represent a small domain in which we conducted our case study. In the first case study, we conducted study on 10 users for 10 usage periods of their smartphone battery. As the focus of our study is on understanding of how the user utility can be maximized, subject to the remaining energy in the battery and the user preferences for allocating the battery energy for the types of application uses, our findings are fundamentally tied to reflect this objective by setting the values of the parameters of our proposed model rather than investigating smartphone and application usage patterns such as frequency, duration, types of interactions and etc. Although the results of this study were based on small sample of users, however, there is a clear value in conducting such study in understanding the significant of the user preference in optimizing the energy consumption. In addition, such case studies provide insights about human-mobile interaction for future mobile development.

For the second study, usually experiments with Neural Networks require large amount of training data . However, in this work we just used a dataset of 2000 patterns labeled. Therefore, in the future work the dataset can be increased with much more labels from more than one user. Also, the offloading decision made by the network can be used to extend the learning data set and enhance the network performance.

Bibliography

Abolfazli, S., Sanaei, Z., Ahmed, E., Gani, A., & Buyya, R. (2014). Cloud-based augmentation for mobile devices: motivation, taxonomies, and open challenges. *Communications Surveys & Tutorials, IEEE, 16*(1), 337-368.

Abrahamse, W., Steg, L., Vlek, C., & Rothengatter, T. (2007). The effect of tailored information, goal setting, and tailored feedback on household energy use, energy-related behaviors, and behavioral antecedents. *Journal of environmental psychology, 27(4), 265-276.*

Ahmed, E., Gani, A., Khan, M. K., Buyya, R., & Khan, S. U. (2015). Seamless application execution in mobile cloud computing: Motivation, taxonomy, and open challenges. *Journal of Network and Computer Applications, 52*, 154-172.

Ahmed, E., Gani, A., Sookhak, M., Ab Hamid, S. H., & Xia, F. (2015). Application optimization in mobile cloud computing: Motivation, taxonomies, and open challenges. *Journal of Network and Computer Applications, 52*, 52-68.

Allen, W. B., Doherty, N.,Mansfield, K. W. E., . (2005). *Managerial Economics: Theory, Applications, and Cases*: Norton.

Altamimi, M., Palit, R., Naik, K., & Nayak, A. (2012). Energy-as-a-Service (EaaS): On the efficacy of multimedia cloud computing to save smartphone energy. I*n proc. 2012 Cloud Computing (CLOUD), 2012 IEEE 5th International Conference, 764-771.*

Altmann, J., & Chu, K. (2001). A proposal for a flexible service plan that is attractive to users and internet service providers. *In INFOCOM 2001. Twentieth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE, 2, 953-958.*

Altmann, J., & Rohitratana, J. (2010). Software resource management considering the interrelation between explicit cost, energy consumption, and implicit cost. *Multikonferenz Wirtschaftsinformatik 2010*, 71.

Altmann, J., Rupp, B., & Varaiya, P. (1999). Internet demand under different pricing schemes. *In Proc. 1st ACM conference on Electronic commerce, ACM, 9-14*

Altmann, J., Rupp, B., & Varaiya, P. (2001). Effects of pricing on Internet user behavior. *NetNomics, 3*(1), 67-84.

Altmann, J., & Varaiya, P. (1998). *I*NDEX project: user support for buying QoS with regard to user's preferences. *In Quality of Service, 1998.(IWQoS 98) 1998 Sixth International Workshop, IEEE, 101-104.*

Anand, A., Manikopoulos, C., Jones, Q., & Borcea, C. (2007). A quantitative analysis of power consumption for location-aware applications on smart

phones. *In 2007 IEEE International Symposium on Industrial Electronics (pp. 1986-1991). IEEE.*

Anera, P. (2013). Survey shows battery life to be the single main gripe of today's mobile phone user. *from http://www.phonearena.com/news/Survey-shows-battery-life-to-be-the-single-main-gripe-of-todays-mobile-phone-user_id49818*.

Athukorala, K., Lagerspetz, E., von Kügelgen, M., Jylhä, A., Oliner, A. J., Tarkoma, S., & Jacucci, G. (2014). How carat affects user behavior: implications for mobile battery awareness applications. *In Proceedings of the 32nd annual ACM conference on Human factors in computing systems, ACM, 1029-1038.*

Banerjee, N., Rahmati, A., Corner, M. D., Rollins, S., & Zhong, L. (2007). Users and batteries: interactions and adaptive energy management in mobile systems. *Springer Berlin Heidelberg, 217-234.*

Banovic, N., Brant, C., Mankoff, J., & Dey, A. (2014). ProactiveTasks: the short of mobile device use sessions. *In Proceedings of the 16th international conference on Human-computer interaction with mobile devices & services, ACM, 243-252.*

Brenes, A. (2011). Cobb-douglas Utility Function. *Available at https://www.wiziq.com/tutorial/160571-Cobb-Douglas-Utility-Function.*

Carcangiu, S., Fanni, A., & Montisci, A. (2011). Multi-Objective Optimization Methods Based on Artificial Neural Networks. *INTECH Open Access Publisher. DOI:10.5772/15113.*

Cardellini, V., Personé, V. D. N., Di Valerio, V., Facchinei, F., Grassi, V., Presti, F. L., & Piccialli, V. (2013). A game-theoretic approach to computation offloading in mobile cloud computing. *Mathematical Programming*, 1-29.

Carroll, A., & Heiser, G. (2010). An Analysis of Power Consumption in a Smartphone. *In USENIX annual technical conference, 14.*

Chan, W., Jaitly, N., Le, Q. V., & Vinyals, O. (2016). Listen, Attend and Spell: A Neural Network for Large Vocabulary Conversational Speech Recognition. *In 2016 IEEE International Conference on Acoustics, Speech and signal Processing (ICASSP), IEEE, 4960-4964.*

Chetan, S., Kumar, G., Dinesh, K., Mathew, K., & Abhimanyu, M. Cloud Computing for Mobile World. *Available at chetan. ueuo. com. 2010.*

Christensen, J. H. (2009). *U*sing RESTful web-services and cloud computing to create next generation mobile applications. *In Proceedings of the 24th ACM SIGPLAN conference companion on Object oriented programming systems languages and applications, ACM, 627-634.*

Chun, B.-G., Ihm, S., Maniatis, P., Naik, M., & Patti, A. (2011). Clonecloud: elastic execution between mobile device and cloud. *In Proceedings of the sixth conference on Computer systems, ACM, 301-314.*

Church, K., Ferreira, D., Banovic, N., & Lyons, K. (2015). Understanding the Challenges of Mobile Phone Usage Data. *In Proceedings of the 17th International Conference on Human-Computer Interaction with Mobile Devices and Services, ACM, 504-514.*

Cichy, R. M., Khosla, A., Pantazis, D., Torralba, A., & Oliva, A. (2016). Deep Neural Networks predict Hierarchical Spatio-temporal Cortical Dynamics of Human Visual Object Recognition. *arXiv preprint arXiv:1601.02970*.

Ciresan, D. C., Meier, U., Gambardella, L. M., & Schmidhuber, J. (2010). Deep, big, simple neural nets for handwritten digit recognition. *Neural computation, 22*(12), 3207-3220.

Cireşan, D. C., Meier, U., Masci, J., Gambardella, L. M., & Schmidhuber, J. (2011). High-performance neural networks for visual object classification. *arXiv preprint arXiv:1102.0183*.

Creus, G. B., & Kuulusa, M. (2007). Optimizing mobile software with built-in power profiling. *Mobile Phone Programming, Springer,* 449-462).

Cuervo, E., Balasubramanian, A., Cho, D.-k., Wolman, A., Saroiu, S., Chandra, R., & Bahl, P. (2010). MAUI: making smartphones last longer with code offload. *In Proceedings of the 8th international*

conference on Mobile systems, applications, and services, ACM, 49-62.

Dahl, G. E., Yu, D., Deng, L., & Acero, A. (2012). Context-dependent pre-trained deep neural networks for large-vocabulary speech recognition. *Audio, Speech, and Language Processing, IEEE Transactions on, 20*(1), 30-42.

Demumieux, R., & Losquin, P. (2005). Gather customer's real usage on mobile phones. *In Proceedings of the 7th international conference on Human computer interaction with mobile devices & services, ACM, 267-270.*

Deng, L., & Togneri, R. (2015). Deep dynamic models for learning hidden representations of speech features. *Speech and Audio Processing for Coding, Enhancement and Recognition, Springer, 153-195.*

Dinh, H. T., Lee, C., Niyato, D., & Wang, P. (2011). A Survey of Mobile Cloud Computing: Architecture. *Applications, and Approaches*, 1-25.

Dinh, H. T., Lee, C., Niyato, D., & Wang, P. (2013). A survey of mobile cloud computing: architecture, applications, and approaches. *Wireless communications and mobile computing, 13*(18), 1587-1611.

Do, T. M. T., Blom, J., & Gatica-Perez, D. (2011). Smartphone usage in the wild: a large-scale analysis of applications and context. *Paper presented at the Proceedings of the 13th international conference on multimodal interfaces, ACM, 353-360.*

Donahue, J., Jia, Y., Vinyals, O., Hoffman, J., Zhang, N., Tzeng, E., & Darrell, T. (2013). Decaf: A deep convolutional activation feature for generic visual recognition. *arXiv preprint arXiv:1310.1531*.

Enzai, M., Idawati, N., & Tang, M. (2014). A taxonomy of computation offloading in mobile cloud computing. *In Mobile Cloud Computing, Services, and Engineering (MobileCloud), 2014 2nd IEEE International Conference on. 2014, IEEE, 19-28.*

Falaki, H., Lymberopoulos, D., Mahajan, R., Kandula, S., & Estrin, D. (2010). A first look at traffic on smartphones. *In Proceedings of the 10th ACM SIGCOMM conference on Internet measurement, ACM, 281-287.*

Falaki, H., Mahajan, R., Kandula, S., Lymberopoulos, D., Govindan, R., & Estrin, D. (2010). Diversity in smartphone usage. *In Proceedings of the 8th international conference on Mobile systems, applications, and services, ACM, 179-194.*

Feeney, L. M., & Nilsson, M. (2001). Investigating the energy consumption of a wireless network interface in an ad hoc networking environment. *In INFOCOM 2001. Twentieth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE, 1548-1557.*

Fernando, N., Loke, S. W., & Rahayu, W. (2013). Mobile cloud computing: A survey. *Future Generation Computer Systems, 29*(1), 84-106.

Ferreira, D., Dey, A. K., & Kostakos, V. (2011). Understanding human-smartphone concerns: a study of battery life *Pervasive computing* , Springer, 19-33.

Ferreira, D., Ferreira, E., Goncalves, J., Kostakos, V., & Dey, A. K. (2013). Revisiting human-battery interaction with an interactive battery interface. *In Proceedings of the 2013 ACM international joint conference on Pervasive and ubiquitous computing, ACM, 563-572.*

Ferreira, D., Goncalves, J., Kostakos, V., Barkhuus, L., & Dey, A. K. (2014). Contextual experience sampling of mobile application micro-usage. *In Proceedings of the 16th international conference on Human-computer interaction with mobile devices & services, ACM, 91-100.*

Froehlich, J., Chen, M. Y., Consolvo, S., Harrison, B., & Landay, J. A. (2007). MyExperience: a system for in situ tracing and capturing of user feedback on mobile phones. *In Proceedings of the 5th international conference on Mobile systems, applications and services, ACM, 57-70.*

Ghasemi-Falavarjani, S., Nematbakhsh, M., & Ghahfarokhi, B. S. (2015). Context-aware multi-objective resource allocation in mobile cloud. *Computers & Electrical Engineering, 44,* 218-240.

Giurgiu, I., Riva, O., Juric, D., Krivulev, I., & Alonso, G. (2009). Calling the cloud: enabling mobile phones as interfaces to cloud applications *Middleware 2009* , Springer, 83-102).

Gupta, A., & Mohapatra, P. (2007). Energy consumption and conservation in wifi based phones: A measurement-based study. In 2007 4th Annual IEEE Communications Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks (pp. 122-131). IEEE.

Haile, N., & Altmann, J. (2013). Estimating the value obtained from using a software service platform *Economics of Grids, Clouds, Systems, and Services* , Springer, 244-255.

Haile, N., & Altmann, J. (2016). Structural analysis of value creation in software service platforms. *Electronic Markets*, 1-14.

Hasan, M. Z., Kamil, A. A., Mustafa, A., & Baten, M. A. (2012). A Cobb Douglas stochastic frontier model on measuring domestic bank efficiency in Malaysia. *PLOS one, 7*(8), e42215.

Hayes, R. M. (1983). An Application of the Cobb-Douglas Model to the Association of Research Libraries. *Library and Information Science Research, An International Journal, 5*(3), 291-325.

Haykin, S., & Network, N. (2004). A comprehensive foundation. *Neural Networks, 2*(2004).

Heikkinen, M. V., Nurminen, J. K., Smura, T., & HäMmäInen, H. (2012). Energy efficiency of mobile handsets: Measuring user attitudes and behavior. *Telematics and Informatics, 29*(4), 387-399.

Hinton, G., Deng, L., Yu, D., Dahl, G. E., Mohamed, A.-r., Jaitly, N., . . . Sainath, T. N. (2012). Deep neural networks for acoustic modeling in

speech recognition: The shared views of four research groups. *Signal Processing Magazine, IEEE, 29*(6), 82-97.

Huerta-Canepa, G., & Lee, D. (2010). A virtual cloud computing provider for mobile devices. *In Proceedings of the 1st ACM Workshop on Mobile Cloud Computing & Services: Social Networks and Beyond, ACM, 6.*

Hung, S.-H., Shih, C.-S., Shieh, J.-P., Lee, C.-P., & Huang, Y.-H. (2012). Executing mobile applications on the cloud: Framework and issues. *Computers & Mathematics with Applications, 63*(2), 573-587.

ICT facts & figures, T. W. i. (2015). *from https://www.itu.int/en/ITU-D/Statistics/Documents/facts/ICTFactsFigures2015.pdf*

IDC. (2015). Smartphone OS Market Share, 2015 Q., *from http://www.idc.com/prodserv/smartphone-os-market-share.js*

Jarrett, K., Kavukcuoglu, K., Ranzato, M. A., & LeCun, Y. (2009). What is the best multi-stage architecture for object recognition? *In IEEE 12th International Conference on Computer vesion, IEEE, 2146-2153.*

Jesdabodi, C., & Maalej, W. (2015). Understanding usage states on mobile devices. *In Proceedings of the 2015 ACM International Joint Conference on Pervasive and Ubiquitous Computing, ACM, 1221-1225.*

Jones, S. L., Ferreira, D., Hosio, S., Goncalves, J., & Kostakos, V. (2015). Revisitation analysis of smartphone app use. *In Proceedings of the*

2015 ACM International Joint Conference on Pervasive and Ubiquitous Computing, ACM, 1197-1208.

Kang, J.-M., Park, C.-K., Seo, S.-S., Choi, M.-J., & Hong, J. W.-K. (2008). User-centric prediction for battery lifetime of mobile devices. *Challenges for Next Generation Network Operations and Service Management*, Springer, 531-534.

Kang, J.-M., Seo, S.-s., & Hong, J. W.-K. (2011). Usage pattern analysis of smartphones. *In Network Operations and Management Symposium (APNOMS), 2011 13th Asia-Pacific, IEEE, 1-8.*

Karamshuk, D., Boldrini, C., Conti, M., & Passarella, A. (2011). Human mobility models for opportunistic networks. *Communications Magazine, IEEE, 49*(12), 157-165.

Kemp, R., Palmer, N., Kielmann, T., & Bal, H. (2010). Cuckoo: a computation offloading framework for smartphones *Mobile Computing, Applications, and Services* , Sprinfger, 59-79.

Kim, J., Ilon, L., & Altmann, J. (2013). Adapting smartphones as learning technology in a Korean university. *Journal of Integrated Design and Process Science, 17*(1), 5-16.

Kosta, S., Aucinas, A., Hui, P., Mortier, R., & Zhang, X. (2012). Thinkair: Dynamic resource allocation and parallel execution in the cloud for mobile code offloading. *In INFOCOM, 2012 Proceedings IEEE, 945-953.*

Koukoumidis, E., Lymberopoulos, D., Strauss, K., Liu, J., & Burger, D. (2011). Pocket cloudlets. *ACM SIGARCH Computer Architecture News, 39*(1), 171-184.

Kovachev, D., Yu, T., & Klamma, R. (2012). Adaptive computation offloading from mobile devices into the cloud. *In 2012 IEEE 10th International Symposium on. Parallel and Distributed Processing with Applications (ISPA), IEEE, 784-791*.

Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. *In Advances in neural information processing systems, 1097-1105*.

Kumar, K., Liu, J., Lu, Y.-H., & Bhargava, B. (2013). A survey of computation offloading for mobile systems. *Mobile Networks and Applications, 18*(1), 129-140.

Kumar, K., & Lu, Y.-H. (2010). Cloud computing for mobile users: Can offloading computation save energy? *Computer*(4), 51-56.

Lim, S. L., Bentley, P. J., Kanakam, N., Ishikawa, F., & Honiden, S. (2015). Investigating country differences in mobile app user behavior and challenges for software engineering. *Software Engineering, IEEE Transactions on, 41*(1), 40-64.

Lin, T.-Y., Lin, T.-A., Hsu, C.-H., & King, C.-T. (2013). Context-aware decision engine for mobile cloud offloading. *In Wireless*

Communications and Networking Conference Workshops (WCNCW), IEEE, 111-116.

Liu, L., Moulic, R., & Shea, D. (2010). Cloud service portal for mobile device management. *In 2010 IEEE 7th International Conference on e-Business Engineering (ICEBE), IEEE, 474-478.*

Magurawalage, C. M. S., Yang, K., Hu, L., & Zhang, J. (2014). Energy-efficient and network-aware offloading algorithm for mobile cloud computing. *Computer Networks, 74*, 22-33.

Marinelli, E. E. (2009). Hyrax: cloud computing on mobile devices using MapReduce. *DTIC Document.*

McCalley, L. T., & Midden, C. J. (2002). Energy conservation through product-integrated feedback: The roles of goal-setting and social orientation. *Journal of economic psychology, 23(5), 589-603.*

McMillan, D., Morrison, A., Brown, O., Hall, M., & Chalmers, M. (2010). Further into the wild: Running worldwide trials of mobile systems *Pervasive Computing*, Springer, 210-227.

Miettinen, A. P., & Nurminen, J. K. (2010). Energy Efficiency of Mobile Clients in Cloud Computing. *HotCloud, 10*, 4-4.

Mobifroge. (2014). Global mobile statistics 2014 Part A: Mobile subscribers; handset market share; mobile operators. . *from http://mobiforge.com/research-analysis/global-mobile-statistics-*

2014-part-a-mobile-subscribers-handset-market-share-mobile-operators

Mohamed, A.-r., Dahl, G., & Hinton, G. (2009). Deep belief networks for phone recognition. *In Nips workshop on deep learning for speech recognition and related applications, 1(9), 39.*

Oliner, A. J., Iyer, A. P., Stoica, I., Lagerspetz, E., & Tarkoma, S. (2013). Carat: Collaborative energy diagnosis for mobile devices. *In 11th ACM Conference on Embedded Networked Sensor Systems.*

Oliver, E. (2010). The challenges in large-scale smartphone user studies. *In Proceedings of the 2nd ACM International Workshop on Hot Topics in Planet-scale Measurement, ACM, 10.*

Oliver, E. A., & Keshav, S. (2011). An empirical approach to smartphone energy level prediction. *In Proceedings of the 13th international conference on Ubiquitous computing, ACM, 345-354.*

Ou, S., Yang, K., & Hu, L. (2007). Cross: a combined routing and surrogate selection algorithm for pervasive service offloading in mobile ad hoc environments. *In Global Telecommunications Conference, 2007. GLOBECOM'07., IEEE, 720-725.*

Pal, S. (2015). Extending Mobile Cloud Platforms Using Opportunistic Networks: Survey, Classification and Open Issues. *Journal of Universal Computer Science, 21*(12), 1594-1634.

Pasricha, S., Donohoo, B. K., & Ohlsen, C. (2015). A middleware framework for application-aware and user-specific energy optimization in smart mobile devices. *Pervasive and Mobile Computing, 20*, 47-63.

Pathak, A., Hu, Y. C., & Zhang, M. (2012). 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, ACM, 29-42.*

Peltonen, E., Lagerspetz, E., Nurmi, P., & Tarkoma, S. (2016). Where Has My Battery Gone?: A Novel Crowdsourced Solution for Characterizing Energy Consumption. *Pervasive Computing, IEEE, 15*(1), 6-9.

Principe, J. C., Euliano, N. R., & Lefebvre, W. C. (1999). Neural and adaptive systems: fundamentals through simulations with CD-ROM. *John Wiley & Sons, Inc.*

Rahimi, M. R., Venkatasubramanian, N., Mehrotra, S., & Vasilakos, A. V. (2013). On optimal and fair service allocation in mobile cloud computing. *arXiv preprint arXiv:1308.4391*.

Rahmati, A., Qian, A., & Zhong, L. (2007). Understanding human-battery interaction on mobile phones. *In Proceedings of the 9th international conference on Human computer interaction with mobile devices and services, ACM, 265-272.*

Rahmati, A., Tossell, C., Shepard, C., Kortum, P., & Zhong, L. (2012). Exploring iPhone usage: the influence of socioeconomic differences on smartphone adoption, usage and usability. *In Proceedings of the 14th international conference on Human-computer interaction with mobile devices and services, ACM, 11-20.*

Rahmati, A., & Zhong, L. (2009). Human–battery interaction on mobile phones. *Pervasive and Mobile Computing, 5*(5), 465-477.

Rahmati, A., & Zhong, L. (2010). A longitudinal study of non-voice mobile phone usage by teens from an underserved urban community. *arXiv preprint arXiv:1012.2832*.

Rahmati, A., & Zhong, L. (2013). Studying smartphone usage: Lessons from a four-month field study. *Mobile Computing, IEEE Transactions on, 12*(7), 1417-1427.

Ravi, A., & Peddoju, S. K. (2015). Handoff strategy for improving energy efficiency and cloud service availability for mobile devices. *Wireless Personal Communications, 81*(1), 101-132.

Ravi, N., Scott, J., Han, L., & Iftode, L. (2008). Context-aware battery management for mobile phones. *In Pervasive Computing and Communications, 2008. PerCom 2008. Sixth Annual IEEE International Conference on Prevasive Computing and Communications, IEEE, 224-233.*

Robinson, S. (2009). Cellphone energy gap: desperately seeking solutions, Strategy Analytics, *(2009).*

Rumelhart, D. E., Hinton, G. E., & Williams, R. J. (1988). Learning representations by back-propagating errors. *Cognitive modeling, 5(3), 1*.

Russell, S. J., Norvig, P., Canny, J. F., Malik, J. M., & Edwards, D. D. (2003). Artificial intelligence: a modern approach *Upper Saddle River: Prentice hall, (Vol. 2).*

Satyanarayanan, M., Bahl, P., Caceres, R., & Davies, N. (2009). The case for vm-based cloudlets in mobile computing. *Pervasive Computing, IEEE, 8(4), 14-23.*

Seide, F., Li, G., & Yu, D. (2011). Conversational Speech Transcription Using Context-Dependent Deep Neural Networks. *In Interspeech, 437-440.*

Shah, S. A. A., Bennamoun, M., & Boussaid, F. (2016). Iterative deep learning for image set based face and object recognition. *Neurocomputing, 174*, 866-874.

Shikfa, A. (2010). Security issues in opportunistic networks. *In Proceedings of the Second International Workshop on Mobile Opportunistic Networking, ACM, 215-216.*

Shiraz, M., Gani, A., Khokhar, R. H., & Buyya, R. (2013). A review on distributed application processing frameworks in smart mobile

devices for mobile cloud computing. *Communications Surveys & Tutorials, IEEE, 15*(3), 1294-1313.

Shiraz, M., Sookhak, M., Gani, A., & Shah, S. A. A. (2015). A Study on the Critical Analysis of Computational Offloading Frameworks for Mobile Cloud Computing. *Journal of Network and Computer Applications, 47*, 47-60.

Shye, A., Scholbrock, B., & Memik, G. (2009). Into the wild: studying real user activity patterns to guide power optimizations for mobile architectures. *In Proceedings of the 42nd Annual IEEE/ACM International Symposium on Microarchitecture, ACM, 168-178.*

Simonyan, K., & Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556.*

Soyata, T., Ba, H., Heinzelman, W., Kwon, M., & Shi, J. (2013). Accelerating mobile cloud computing: A survey. *Communication Infrastructures for Cloud Computing*, 175-197.

Statista. (2011). Consumer satisfaction with battery life of wireless devices in the United States in 2011 (in percent). . *from* *http://www.statista.com/statistics/216130/satisfaction-with-battery-life-of-mobile-devices-among-us-consumers/*

Statista. (2015). Number of apps available in leading app stores as of July 2015., *from* *http://www.statista.com/statistics/276623/number-of-apps-available-in-leading-app-stores/*

Tarkoma, S., Siekkinen, M., Lagerspetz, E., & Xiao, Y. (2014). Smartphone energy consumption: modeling and optimization. *Cambridge University Press.*

Vallerio, K. S., Zhong, L., & Jha, N. K. (2006). Energy-efficient graphical user interface design. *IEEE Transactions on Mobile Computing, 5(7), 846-859.*

Vallina-Rodriguez, N., Hui, P., Crowcroft, J., & Rice, A. (2010). Exhausting battery statistics: understanding the energy demands on mobile handsets. *In Proceedings of the second ACM SIGCOMM workshop on Networking, systems, and applications on mobile handhelds, ACM, 9-14.*

Varian, H. R. (2014). Intermediate Microeconomics: A Modern Approach. *Ninth International Student Edition*: *WW Norton & Company.*

Wei, X., Fan, J., Lu, Z., & Ding, K. (2013). Application scheduling in mobile cloud computing with load balancing. *Journal of Applied Mathematics, 2013*.

Wen, Y., Zhang, W., & Luo, H. (2012). Energy-optimal mobile application execution: Taming resource-poor mobile devices with cloud clones. *In INFOCOM, 2012 Proceedings, IEEE, 2716-2720.*

Wolski, R., Gurun, S., Krintz, C., & Nurmi, D. (2008). Using bandwidth data to make computation offloading decisions. *In IPDPS 2008, IEEE*

International Symposium on Parallel and Distributed Processing, IEEE, 1-8.

Wu, H., Huang, D., & Bouzefrane, S. (2013, October). Making offloading decisions resistant to network unavailability for mobile cloud collaboration. *In, 2013 9th International Conference Conference on Collaborative Computing: Networking, Applications and Worksharing (Collaboratecom), IEEE , 168-177.*

Xia, F., Ding, F., Li, J., Kong, X., Yang, L. T., & Ma, J. (2014). Phone2Cloud: Exploiting computation offloading for energy saving on smartphones in mobile cloud computing. *Information Systems Frontiers, 16*(1), 95-111.

Xiao, Y., Kalyanaraman, R. S., & Yla-Jaaski, A. (2008, September). Energy consumption of mobile youtube: Quantitative measurement and analysis. *In 2008 the Second International Conference on Next Generation Mobile Applications, Services, and Technologies, IEEE, . 61-69).*

Yan, B., Shi, F., & Yu, R.-q. (2013). Exploring utility function in utility management: an evaluating method of library preservation. *SpringerPlus, 2*(1), 1-11.

Zhang, L., Tiwana, B., Qian, Z., Wang, Z., Dick, R. P., Mao, Z. M., & Yang, L. (2010). Accurate online power estimation and automatic battery behavior based power model generation for smartphones. *In*

Proceedings of the eighth IEEE/ACM/IFIP international conference on Hardware/software codesign and system synthesis, ACM, 105-114.

Zhang, X., Kunjithapatham, A., Jeong, S., & Gibbs, S. (2011). Towards an elastic application model for augmenting the computing capabilities of mobile devices with cloud computing. *Mobile Networks and Applications, 16*(3), 270-284.

Zhao, X., Guo, Y., Feng, Q., & Chen, X. (2011). A system context-aware approach for battery lifetime prediction in smart phones. *In Proceedings of the 2011 ACM Symposium on Applied Computing, ACM, 641-646.*

Zhou, B., Dastjerdi, A. V., Calheiros, R. N., Srirama, S. N., & Buyya, R. (2015). A context sensitive offloading scheme for mobile cloud computing service. *In 2015 IEEE 8th International Conference on Cloud Computin, IEEE, 869-876.*

Zhou, B., Vahid Dastjerdi, A., Calheiros, R., Srirama, S., & Buyya, R. (2015). mCloud: A Context-aware Offloading Framework for Heterogeneous Mobile Cloud. *IEEE, DOI 10.1109/TSC.2015.2511002, IEEE Transaction on Service Computing.*

Zhou, P., Jiang, H., Dai, L.-R., Hu, Y., & Liu, Q.-F. (2015). State-clustering based multiple deep neural networks modeling approach for speech recognition. *Audio, Speech, and Language Processing, IEEE/ACM Transactions on, 23*(4), 631-642.

국문초록

스마트폰의 자원 제약을 극복하기 위한 사용자 선호 기반 최적화 방법론 연구

스마트폰 기술과 기기의 애플리케이션 개발 분야에서 이루어진 기술적 발전은 지난 몇년간의 스마트폰 사용에 상당한 영향을 미쳤다. 최근에 우리는 스마트폰을 단순한 전화 통화, 문자 송수신 서비스 뿐만 아니라, SNS(social network service), 인터넷, 이메일, 게임, 동영상 시청등의 다양한 서비스를 이용하는데 사용한다. 하지만, 일부 서비스들은 스마트폰의 작동에 무리를 주며, 메모리, 정보처리능력, 인터넷 연결정도, 배터리와 같은 단말기 성능의 엄청난 소모로 이어진다.

스마트폰의 매력적인 디자인, 다양한 기능과 넓은 범위의 활용성에도 불구하고, 스마트폰은 배터리, 정보 처리능력, 네트워크 대역폭, 용량등에 있어서 여전히 한계가 있다. 이러한 휴대폰 성능의 한계는 제한적인 스마트폰 사용으로 나타났고, 한계를 극복하기 위한 해결책에 대한 필요성이 급증하게 되었다.

그 중에서도 스마트폰의 배터리 수명은 가장 중요한 한계점이다. 이는 스마트폰 사용자들 뿐만 아니라, 제조사들에게도 항상 신경쓰이는 문제이다. 배터리 기술 향상을 위해 상당한 노력을 기울였음에도 불구하고, 배터리 기술

진보는 휴대폰 사용을 위해 필요한 배터리 용량에 대한 사용자들의 요구를 따라가지 못하고 있다.

스마트폰 배터리 수명은 사용자의 행위에 달려있기 때문에, 사용자가 어떻게 스마트폰의 배터리를 소모하는지 이해할 필요가 있다. 많은 연구자들이 스마트폰 사용자의 사용행태를 분석하기 위해 노력했다. 이들은 대부분의 사용자가 자신만의 사용 유형이 있으며, 사용자들의 행위를 이해하는 것이 중요함을 보여줬다. 사용자 행동에 대한 이해 없이는 모바일 전력 소비 최적화 혹은 사용자 경험의 최적화를 위한 요인을 명확히 아는 것이 불가능하다.

최근에 모바일 클라우드 컴퓨팅(Mobile Cloud Computing, MCC)이 스마트폰의 한계를 극복하기위하여 소개되었다. 모바일 클라우드 컴퓨팅은 성능 소모가 큰 애플리케이션을 사용할 때, 단말기 대신에 기기보다 사양이 높은 클라우드상에서 구동할 수 있도록 해준다. 이를 컴퓨팅 오프로딩(computing offloading) 이라고 한다. 하지만, 클라우드를 활용한 오프로딩은 사용자가 이용한만큼 비용을 지불해야 하기 때문에 단말기에서 사용하는 비용보다 오프로딩 비용이 더 적을때만 사용하게 된다. 게다가 클라우드상에서 애플리케이션을 오프로드로 실행할 때, 단말기의 배터리 상태나 사용자의 위치와 같은 단말기 사용환경에 따라 오프로딩의 사용편의도가 달라진다.

이번 연구의 목적은 에너지 소비와 스마트폰 애플리케이션 실행에 있어서, 기기의 성능을 제한하는 스마트폰의 한계점들을 극복할 수 있는

해결책을 찾는 것이다. 스마트폰의 성능에 대한 제한점들을 설명하기 위해서 본 논문에서는 2 개의 주요 문제에 초점을 맞춘다. 첫째, 배터리 사용에 대한 에너지 소비의 최적화 문제이다. 이는 스마트폰 배터리에 남아있는 에너지에서 사용자 효용을 최대화하기 위한 사용자 선호도를 고려하여 분석하였다. 특히, 단말기상의 애플리케이션 사용과 클라우드상의 애플리케이션 사용의 2 가지 유형의 스마트폰 애플리케이션 사용 방식을 제시한다. 그리고, 배터리 잔량을 유형별로 어떻게 배분할것인지에 대한 사용자 선호도를 고려하여, 사용자효용기반 에너지 소비 최적화 모형(a utility-based energy consumption optimisation model)을 제안한다. 제안된 모형은 실제 스마트폰 사용자들의 사용 데이터와 대면 인터뷰를 통하여 수집한 자료의 정량적 분석 및 정성적 분석에 활용된다.

둘째, 모바일 클라우드 컴퓨팅에 스마트폰 애플리케이션의 오프로딩을 지원하는 다양한 범주의 최적화 방법에 대해 분석한다. 여기서 제안되는 오프로딩 결정 모형 (an offloading decision making model)은 사용자 환경과 선호도를 반영하여 애플리케이션, 스마트폰, 네트워크, 클라우드 대한 한계점들을 포함하는 다중적 제약이 문제가 되는 오프로딩 비용에 대하여, 이를 최소화하기 위한 결정을 분석한다. 애플리케이션을 단말기상에서 구동할 것인지 혹은 클라우드상에서 구동할 것인지에 따라 비용이 다르게 발생한다. 이러한 각각의 비용들은 다양한 범주의 사용자의 선호에 따라 통합된다.

사용자들이 다양한 요인들을 고려하여 좋은 결정을 내린다고 가정할 수 있지만, 빈번하게 그러한 결정을 내려야하는 것은 번거로울 것이다. 더욱이

오프로딩 결정에 영향을 미칠 수 있는 모든 요인들간의 관계를 설명하는 특정한 선형함수가 없다. 따라서, 다양한 요인들사이의 비선형 관계를 설명하는 모형을 만들기 위하여 신경망(neural network)을 연구에 사용하였다. 이와 관련하여, 심층 신경망(Deep Neural Network, DNN)은 장기적으로 사용자가 최적의 오프로딩 결정을 내릴 수 있도록 지원하고자 한다. 이를 위하여, 사용자에 의해 만들어진 오프로딩 결정 예시들이 앞에서 제안된 오프로딩 결정 모형에 기반하여 설정되었다. 또한, 오프로딩 결정에 대한 가능한 신경망은 사용자 사례(use case)를 통하여 평가되었다.

이 연구는 스마트폰의 성능을 제한하는 한계점들을 극복하기 위한 새로운 접근법들을 제안한다. 스마트폰 사용과 에너지 소비, 컴퓨팅 오프로딩에 대한 이전의 연구들은 대부분 기술적 해결책의 개발에 집중하며 사용자들은 거의 고려하지 않는다. 그러한 기술적 해결책들은 사용자 의도와 선호를 반영하지 못하기 때문에, 스마트폰 사용에 대한 사용자의 기대를 맞추는데 실패할 수 밖에 없다. 따라서, 이 연구는 더욱 향상된 스마트폰 기술 분야에 기여할 수 있는 새로운 연구분야에 대한 시작점이 될 것이라고 생각한다.

주제어: 스마트폰, 스마트폰 앱, 배터리 소비 최적화, 효용함수, 사용 행태, 클라우드 컴퓨팅, 애플리케이션 분류, 에너지 분배, 컴퓨팅 오프로딩, 모바일 클라우드 컴퓨팅, 비용 모형, 사용자 선호, 오프로딩 결정 모형, 신경망, 최적화

학번 : 2012-31306