

A NEW CONTEXT-SENSITIVE DECISION MAKING SYSTEM FOR MOBILE CLOUD OFFLOADING

Mustafa Tanrıverdi¹ and M. Ali Akcayol²

¹Institute of Information, Gazi University, Ankara, Turkey

²Department of Computer Engineering, Gazi University, Ankara, Turkey

ABSTRACT

Recently, with the rapid spread use of mobile devices, some problems have begun to emerge. The most important of these are that the mobile devices batteries' life may be short and that these devices may be in some cases. The complex tasks that must be addressed to solve such problems on mobile devices can be transferred to the cloud environment when appropriate conditions are met. The decision to offload to the cloud environment at this stage is very important. In this thesis, a context-aware decision-making system has been developed for offloading to cloud environments. Unlike similar tasks, the processes determined for transfer to the cloud are not run randomly, but rather according to the mobile user's application usage habits. The developed system was implemented in a real environment for one month. According to the results, it was determined that processes transferred to the cloud were completed in less time and consumed less energy.

KEYWORDS

Mobile Cloud Offloading, Mobile Cloud Computing, Context-Aware System, Forecasting, Dynamic Estimation, Energy-Efficiency

1. INTRODUCTION

Mobile device sales have been increasing rapidly in recent years [1]. These devices can run diverse applications to meet billions of users' requests. Examples of these applications include advanced media players, GPS applications, real-time gaming applications, and communications applications. These mobile applications typically require user interaction, intensive data processing, fast response times, and long battery life. When compared to computers, however, mobile phones have more limited features in terms of processor capacity and battery life. Running complex applications on a mobile device can lead to poor performance and shorten its battery life [2]. Cloud computing services have become increasingly popular in recent years at low cost, along with scalable and flexible architecture. The most appropriate way to overcome the problems caused by the limitations of mobile devices can be thought of as mobile cloud offloading [3].

Transferring a process to the cloud and retrieving the results naturally requires time and energy. At the same time, the capacity and cost of the cloud environment must be taken into consideration. A decision must be made to transfer the process to the cloud, taking into account the cost of the transfer [2].

Many studies have been done on transferring a process from mobile devices to the cloud environment [2, 4–8]. Unlike other studies, this study focuses on two main issues. The first is the integration of the PowerTutor (PT) application into the developed application in order to be able to measure power on the mobile device instantly without the need for any additional hardware. The second is the execution of complex operations and web services with the AsyncTask class. Many of the applications in similar studies were developed based on the Android operating system, in which a complex process must be run with the AsyncTask class. This allows a complex process to take longer to complete by allocating fewer resources rather than using all the resources of the mobile device to complete it more quickly. For example, a process that can be completed in 5 seconds using all available resources on the mobile device can be completed in 50 seconds by using fewer resources on the backplane [9]. This makes it more complex to measure the performance of the processes running on the developed mobile cloud application.

When developing a mobile cloud application to run in a real environment, one must consider the requirement to use the AsyncTask class and instant power measurement. In this study, all processes are executed with AsyncTask class. For dynamic power measurement, the PT application is customized and integrated into the developed application.

The complex processes selected for transfer to the cloud in the studies involving the application were chosen to run randomly at certain time intervals. In this study, the selected processes were run according to the usage habits of the mobile user. For this, the applications that mobile users use in their daily life are paired with determined processes to be transferred to the cloud and run in the background simultaneously.

In this research study, the following subjects have been developed and improved:

- Instantaneous power measurement is very important for a dynamic system that decides to offloading to the cloud without needing any additional hardware. In this study, the open-source PT application is integrated into the developed system for instantaneous power measurement and the source code is customized for mobile devices of different brands.
- Due to the working principle of mobile operating systems, a complex process that can negatively affect the operation of the device's main processes is carried out in the background. The measured power value during the execution of a process is very important in the decision-making stages. It may not be very accurate to calculate this measurement using the difference in power between the start and end of the process. Instead, calculating with power values obtained before and during the operation of the relevant process will provide more accurate values. This study is unique in this respect; there is no study on this topic in the literature.
- It has been determined that most studies in this area are conducted in the simulation or experimental environments. The mobile cloud system developed in this study was implemented in a real environment and the results are compared.
- Processes determined for the application are not run at random time intervals as in similar studies, but simultaneously in the background according to the usage habits of the mobile users.
- One major disadvantage of mobile devices is their short battery life. Due to this, the time when the mobile device will be plugged in is estimated, and until this time, information about whether the current battery-power level is adequate is used in the decision stages.

2. RELATED WORK

A survey was conducted by Kumar and others in the field of mobile cloud offloading. According to the data obtained, processes are transferred to the cloud environment in order to achieve high performance or power savings [10]. In a study by Zhou and others, the PowerTutor application has been used for dynamic access to power-consumption data [11]. At the same time, criteria like power consumption, internet bandwidth, cloud resource status, signal level, and GSM internet usage fees are used in making the decision to transfer a process to the cloud. In the study conducted by Magurawalage and others, the power-consumption data and instantaneous internet connection level values were used in the decision-making stages [12]. In addition, the cost values of transferring to, working in, and receiving results from the cloud are also taken into consideration. A special algorithm has been developed using this data and simulated with the CloudSim application. Lin and others have developed a decision engine called CADA for offloading to the cloud [2]. In addition to the performance data of past processes, CADA also uses information like internet connection and location in its decision stages. The decision-making engine developed in the study was implemented in the real environment. Hao and others conducted a study to determine the energy efficient tasks. Then these tasks are provided for caching and offloading. They formulate offloading problem as a mixed integer nonlinear programming and propose efficient algorithm to solve this problem. Simulation result have shown that proposed scheme has low energy cost compared to other schemes [5]. Chun and others have developed an application called CloneCloud in their study [7]. The CloneCloud application allows a process to be transferred to the cloud in parts rather than as a whole. Lim and Lee have conducted a study on the pre-calculation of the energy consumed by processes transferred from mobile to cloud [13]. It is noted that the pre-accurate estimate of the energy consumption is very important in the stage of transferring processes to the cloud. In this study, an algorithm for estimating dynamic energy consumption has been developed. The developed algorithm is compared with decision table, linear regression, local weighted learning, sample based learning and KStar algorithms. These algorithms were compared according to their ability to predict energy consumption consumed as a result of facial recognition, chess, and video processing being transferred to the cloud. In the facial recognition process, the proposed algorithm and the KStar algorithm are close to each other and reach the correct estimates. In chess application, the proposed algorithm made more accurate estimations than the other algorithms. In the video processing application, the proposed algorithm and the linear regression algorithm have obtained the most accurate estimates. Aldmour and others proposed an offloading model contains a decision engine to examine the feasibility of the offloading [14]. This model investigates whether or not a mobile phone preserves energy by offloading to cloud computing. It has been determined that file size is important in this phase. This model takes into account the completion time and the required energy values. According to experimental results, the performance of the cloud is increased as the file size increases. Li and others proposed an energy-aware task offloading mechanism in multiuser mobile cloud computing. Many criteria are used in the decision stages and the solution of the optimization problem is solved by 0-1 nonlinear integer programming. Proposed algorithm compared with competition-based algorithm and user-satisfaction algorithm. The simulation results validate that the proposed algorithm can achieve better performances [15]. Islam and others have done a study on the processing of mass health data for smart cities [16]. The most important issue here is the transfer of patient data to the cloud system in a short time and with less energy spent. For this, cloudlet structures have been established between mobile devices and the public cloud system. Later, the appropriate Cloudlet selection was made with Ant colony optimization. In the experimental study, patient records consisting of texts, documents and pictures assigned to poisson distribution of 10 patients were transferred to the cloud environment. The results are compared with the options of "not to cloud transfer", "selection of appropriate first cloudlet" and "model with genetic algorithm developed by researchers in previous studies". As a

result, developed system has provided the best values in terms of working time and the number of tasks successfully completed.

When the literature is examined, it is observed that the studies done in this area are limited, having gradually increased in recent years. Generally, studies can be categorized as either collecting data on the mobile environment, making decisions about offloading to the cloud, or offloading a process piece-by-piece. Generally, special algorithms have been developed for solving the specified problems [2, 3, 6, 13, 14, 15, 16] . In a few studies, machine-learning algorithms have been used [17-20]. Large parts of developed applications are context-sensitive. Some of the studies were done on power consumption and completion time, some in the context of internet-connection status, and some in the context of location and time. It has been determined that most applications are run in the simulation environment or experimental application environments. In a few publications, the results are evaluated based on the actual application environment [2, 16, 20]. According to the results of the current studies, the developed solutions can be efficient in terms of energy consumption and performance, the Wi-Fi connection environment is generally more efficient than the GSM connection environment, and the cloud environment provides better performance than does the mobile environment [2, 6, 11, 16, 22-26].

3. CONTEXT-SENSITIVE DECISION MAKING SYSTEM FOR MOBILE CLOUD OFFLOADING

The system comprises two applications: a mobile application that works on mobile devices and a Java web application that works in the cloud environment.

3.1. Mobile Application

The mobile application was developed with Android SDK (Software Development Kit) in the Eclipse application development environment. SQLite was used as the database [27]. The PowerTutor application was configured and integrated into the application. All processes were executed with AsyncTask class [9].

The mobile application has two important modules: the profile-information collection module and the past-process records module.

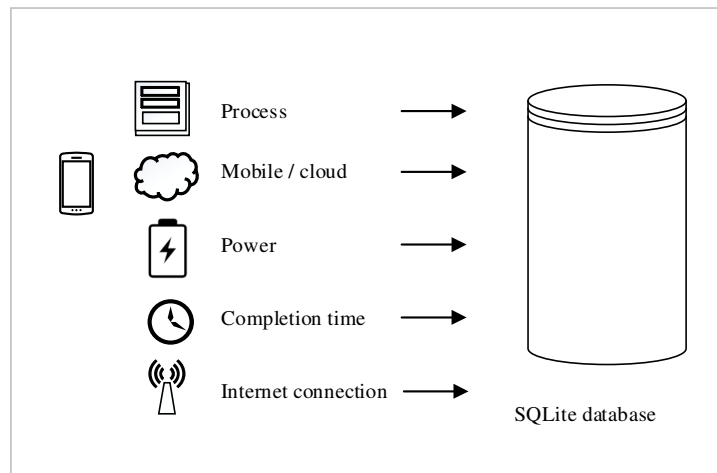


Figure 1. General structure of the past process records module.

Fig. 1 shows the general structure of the past-process records module. The result of running a process in the mobile or cloud environment is recorded in the database. The recorded information is as follows:

- For the process field, the "ID" value of the executed process is recorded.
- For the environment field, the value is set to "1" if the process is run in the mobile environment and "2" if the process is run in the cloud environment.
- The time elapsed during the completion of the process is recorded in the time field.
- For the connection type field, "0" is recorded if the mobile device does not have internet access, "1" if the mobile device's internet connection is provided via Wi-Fi, "2" if it is provided via EDGE, "3" if provided via 3G, and "4" if provided via 4.5G (LTE).
- For the bandwidth field, the internet bandwidth value obtained during the transfer of the relevant process to the cloud is recorded.
- For the power field, the power value calculated during the operation is recorded.

PT application can perform instantaneous power measurement. Rather than calculating the power spent during the execution of a process using the difference between the start and end power values, it is more accurate to obtain the power values at certain intervals during the execution to calculate the average power. At this point, the average power value before the start time should be taken into consideration, along with the average power value measured during the process. In this study, to measure the average power value before the start of the process, the average working time in the past of the relevant process is obtained as t and the time interval between t and the start time of the process is taken into consideration

In Fig. 2, the power values measured for each second are observed before and during a process that takes n seconds to complete. For example, the instantaneous power value at the start of the process is shown as P_0 and the instantaneous power value 1 second after starting is shown as P_1 . The time interval for obtaining instantaneous power value from the data obtained in the PT application is set to 1 second. In other words, the instantaneous power value measured in the PT application can change over at least 1 second.

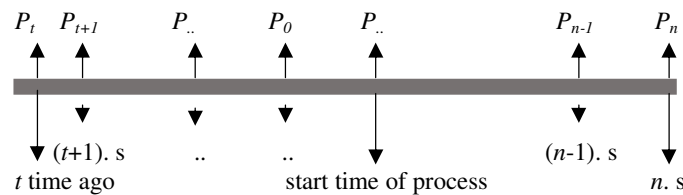


Figure. 2. Calculation of average power value

In equation (1), for a process with an average completion time of t and a working time of n , the difference is calculated between the power mean values before starting and during the work. The resulting P_{ort} value is recorded in the process record table as the average power consumption value.

$$P_{ort} = \frac{\sum_{i=0}^n P_n}{n} - \frac{\sum_{i=0}^t P_{(t+i)}}{t} \quad (1)$$

These performance data of the process are used to estimate the completion time and energy cost in decision stages. In similar studies, the past process information is recorded and used in the decision stages [6, 11, 29].

3.2. Cloud Application

Cloud application is a web project developed with the Java programming language. This application has two basic tasks: answering web service requests from mobile devices and collecting data from mobile application databases in a central database. As part of the first task, the processes running in the mobile application are also provided for the cloud with the same library and the same code. For this task, the Rest service was used as web service structure between the mobile and cloud applications. In the second task, a Mysql database was created in the cloud environment and mobile application data was transferred to this database via web services.

3.3. Application Matching

When similar studies are examined, it seems that complex processes have been used to transfer to the cloud. The Eight Queens Problem, Character Counting in Log File, Face Detection, and Optical Character Recognition (OCR) were selected for use in this study. These were chosen because these complex processes consume more power and have longer completion times than average. These processes are usually chosen from process types such as image processing and mathematical cycle operations. It is not possible for a mobile device user to use an application that involves these processes on a daily basis. Therefore, in the studies involving the application, the specified processes are run randomly by selecting specific time intervals in the background. Unlike similar studies, in this study the processes are run according to the mobile user's application usage habits. For this, profile information, including the application information that the user ran instantly, was recorded at 5-minute intervals for a week. Applications obtained from all users' profile information are rated as H (high), M (medium), or L (low) according to their resource requirements and listed in Table 1.

Table 1. Applications and resource requirements from user profile information

Application	CPU	Storage	Data Transfer	GPU	Battery Usage
Temple Run	H	M	M	H	H
Bike Racing	H	M	M	H	H
Moto Racing	H	M	M	H	H
Real Racing 3	H	M	M	H	H
Candy Crush	M	M	M	M	M
Bubble Shooter	M	M	M	M	M
Word Game	M	M	M	M	M
AutoCAD 360	H	M	L	H	H
WhatsApp	M	H	M	M	M
Navigation Apps	M	L	L	M	H
Snapchat	H	H	H	H	H
Facebook	M	M	H	M	M
Instagram	M	H	H	H	M
Twitter	M	L	M	L	M
Google Drive	M	H	H	L	M
Forum Apps	L	L	M	L	L
Advertisement Applications	M	L	M	L	M
Shopping Apps	M	L	M	L	M
Finance Apps	L	L	M	L	L
Mobile Banking Applications	L	L	M	L	L
News Apps	M	L	M	M	L
Mail Apps	L	L	L	L	L
Browser Apps	M	L	M	M	L

A study by Pandi and Charaf categorized mobile applications according to their resource requirements. Table 2 shows these categories and resource requirements [30].

Table 2. Application categories and resource requirements

Application Type	CPU	Storage	Internet	GPU
Browser Apps	+++	++	+++	++
Browser + Flash Apps	+	+	++	
Games	+++	++	++	+++
Office Apps	++	+	++	
Image View Apps	++	++	++	
Chat(text) Apps	+		+	
Chat (voice) Apps	++		++	
Offline Media Apps	+++	++	+++	+++
Storage Apps	++	+++	+++	
NS/SSH(remote access) Applications	+		++	

Nugroho has provided information on the resource usage of mobile applications in his study. This researcher has benefited from the application named “Terminal” when determining the resource usage of mobile applications. This application can access the log files of the connected mobile device and monitor the resources used by an application [31]. Within the scope of this study, the resource requirements of the processes used in the cloud transfer stages are rated, taking into account the related research in this area [30, .31]. The processes and resource requirements are shown in Table 3.

Table 3. Processes and resource requirements used in the application

No	Process	CPU	Storage	Data Transfer	GPU	Battery Usage
1	Character counting	M	L	M	L	M
2	Eight puzzle	M	L	L	L	L
3	Face detect	H	M	H	H	H
4	OCR	H	M	M	M	H

Table 4. The mapping table of the types of processes to be used in study with the applications used by the users

User Application	Matched Application
Temple Run	Face detect
Bike Racing	Face detect
Moto Racing	Face detect
Real Racing 3	Face detect
Candy Crush	OCR
Bubble Shooter	OCR
Word Game	OCR
AutoCAD 360	Face detect
WhatsApp	OCR
GPS Apps	Character counting
Snapchat	Face detect
Facebook	OCR
Instagram	Face detect
Twitter	Character counting
Google Drive	OCR
Forum Apps	Character counting
Advertisement Apps	Character counting
Shopping Apps	Character counting
Finance Apps	Eight queens puzzle
Mobile Banking Apps	Character counting
News Apps	Eight queens puzzle
Mail Apps	Eight queens puzzle
Browser Apps	Eight queens puzzle

Taking all the information in this section into account, the applications that mobile users use in their daily life are matched to the process types determined under study. The matches shown in Table 4 are transferred to mobile applications via a web service. This allows the process to run throughout the application in the background according to the specified matches.

4. DECISION MODEL FOR MOBILE CLOUD OFFLOADING

The decision model includes energy cost estimation, determination of the critical battery level and the decision algorithm.

4.1. Energy Cost Estimation for Mobile and Cloud Environments

Cuervo and others have calculated the energy cost estimates for running a process in mobile and cloud environments using the least-squares method and simple linear regression [32]. Lim and Lee conducted a study on energy cost estimation using past-process performance data. At the end of the study, the researchers stated that these estimation calculations could be done with the "linear regression", "decision table" and "k nearest neighbors" methods [13]. In the thesis study, Chang made predictions about completion time and power consumption in the cloud and mobile environment using past performance information for a process. This researcher has benefited from multiple linear-regression methods when making these estimates [33]. Khoda and others have also benefited from past performance data while making prediction about the completion time of a process. Researchers have benefited from statistical regression method when making these predictions [28]. In this study, the past performance data has been accessed and the simple linear regression has been used while estimating the energy cost and completion times of a process.

Table 5 shows the symbols used in the calculation of energy cost and completion time. To calculate the cost of energy and completion time if a p process is run with the bt connection type in the cloud environment, all p process records that have been run with the bt connection type in cloud environment are used. Within the specified n records, the completion time values are obtained as $\{T_1, T_2, \dots, T_n\} \in T$, the average power consumption values are obtained as $\{P_1, P_2, \dots, P_n\} \in P$ and internet bandwidth values are obtained as $\{B_1, B_2, \dots, B_n\} \in B$.

Table 5. Symbols and explanations used in calculation of energy cost and completion time

Symbol	Definition
$TEC_{m,p}$	Energy cost required for p process to run in mobile environment (J)
$TEC_{c,p}$	Energy cost required for p process to run in cloud environment (J)
$T_{m,p}$	Estimated completion time of p process in mobile environment (s)
$T_{c,p}$	Estimated completion time of p process in cloud environment (s)
$P_{m,p}$	Estimated power value of p process in mobile environment (mW)
$P_{c,p}$	Estimated power value of p process in cloud environment (mW)
bt	Instant internet connection type of mobile device
B	Instant internet bandwidth of mobile device (bps)

The estimated completion time $T_{c,p}$ for a p process in the cloud environment is obtained by linear regression analysis using B and T values. Related calculations are shown in equations (2), (3) and (4).

$$b_0 = \frac{(\sum T)(\sum B^2) - (\sum B)(\sum BT)}{n(\sum B^2) - (\sum B)^2} \quad (2)$$

$$b_1 = \frac{n(\sum BT) - (\sum B)(\sum T)}{n(\sum B^2) - (\sum B)^2} \quad (3)$$

$$T_{c,p} = b_0 + b_1 \underline{B} \quad (4)$$

Using linear regression analysis using B and P values obtains the estimated energy cost $P_{c,p}$ in the cloud environment for a p process. Related calculations are shown in equations (5), (6) and (7).

$$b_0 = \frac{(\sum P)(\sum B^2) - (\sum B)(\sum BP)}{n(\sum B^2) - (\sum B)^2} \quad (5)$$

$$b_1 = \frac{n(\sum BP) - (\sum B)(\sum P)}{n(\sum B^2) - (\sum B)^2} \quad (6)$$

$$P_{c,p} = b_0 + b_1 \underline{B} \quad (7)$$

After $T_{c,p}$ and $P_{c,p}$ values are obtained, the estimated energy cost for cloud environment $TEC_{c,p}$ is calculated. In similar studies, the estimated energy cost was calculated by multiplying the power and time values [3, 4, 34].

$$TEC_{c,p} = T_{c,p} * P_{c,p} \quad (8)$$

In order to calculate the cost of the estimated energy and completion time if a p process is run in the mobile environment, all p work records that have been run in mobile environment are used. Within the specified n records, the completion time values are obtained as $\{T_1, T_2, \dots, T_n\} \in T$ and the average power consumption values are obtained as $\{P_1, P_2, \dots, P_n\} \in P$. The $T_{m,p}$ value is determined as the average of T values. $P_{m,p}$ value is also determined as the average of P values. Finally, the estimated energy cost for mobile $TEM_{m,p}$ is calculated by multiplying the power and time values as equation (9).

$$TEC_{m,p} = T_{m,p} * P_{m,p} \quad (9)$$

4.2. Critical Battery Level Estimation

In this section it is estimated whether the mobile phone's battery level will be sufficient for normal usage until the next charging time. Studies on this area have been examined [35, 36]. This information is used in the decision-making stages of running a process in the mobile or cloud environment. Before running a process, the information about whether the battery level is sufficient to last until the next charging time is obtained and the weighting factors in the decision algorithm are changed according to this information.

To obtain information about whether the battery level will be sufficient, it is first necessary to estimate when the mobile device will charge. In the next step, it is estimated that the applications will be executed by considering the past application usage habits of the user until the estimated charging time. Next, the amount of energy required to operate these applications is calculated. Then, taking into account the user's past energy consumption data, the amount of energy that the mobile device needs for the process is calculated. Finally, information on the battery status is obtained by comparing the amount of energy required and the amount of energy the mobile device has.

A prediction of the next charging time is estimated will be made based on the user's past charging times. In the first step, the next charging time will be estimated by looking at the user's past charging times. Later, the previous predictions are compared with the actual charging times and more accurate predictions are made by taking into account these prediction errors. It is determined that an accurate prediction cannot be made if the past charging times obtained are very different from each other or if there are too many past errors. In order to predict when the mobile phone will be charged, the past charging times are obtained. The averages and standard deviations are obtained by the minute conversion of the differences between these times and the

instantaneous time. If the calculated standard deviation value is too high, it is determined that the past charging times of this phone are irregular and cannot make an accurate prediction. In cases where the standard deviation value is small, prediction errors made in the past are also used for prediction operations. For a better understanding of these prediction operations, the charge-estimation scenarios are included in the related thesis study.

After estimating the next charging time, the processes to be run before the next charge will be determined by taking advantage of the user's past application-running habits. The Poisson distribution was used at this stage. The four process types to be used for transfer to the cloud are the Eight Queens puzzle, character counting, OCR, and facial detection. These process types are shown in Fig. 3 as $i1$, $i2$, $i3$ and $i4$ respectively. In the following operations, for each process, the number of runs between $t1$ - $t2$ is determined according to the Poisson distribution.

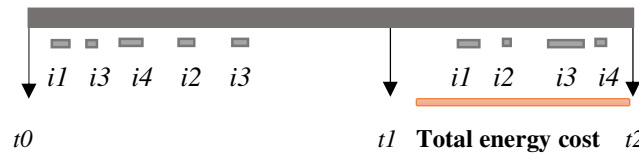


Figure 3. Sample figure for estimation with Poisson distribution.

If an event appropriate to the Poisson distribution occurs λ times in the period $t1-t0$, the probability of its occurrence x times in a different $t2-t1$ period is calculated using equation (10) and equation (11).

$$t = \frac{t1-t0}{t2-t1} \tag{10}$$

$$P(x) = \frac{e^{-\lambda * t} (\lambda * t)^x}{x!}, x = 0, 1, 2, \dots \tag{11}$$

The above operations will result in $x1$, which provides the result of maximum $P(x1)$. The value of $x1$ will be determined as the number of runs for $i1$ between $t1$ and $t2$. The $x2$, $x3$ and $x4$ values are also obtained in similar way.

After the values of $x1$, $x2$, $x3$ and $x4$ are determined, the required energy cost is calculated when the specified processes are run in the mobile environment. The estimated energy cost of running a p process in a mobile environment is expressed as $TEC_{m,p}$. In equation (12), the run number of these processes is multiplied by the $TEC_{m,p}$ values of each process. Then the total energy cost is calculated by gathering the values obtained for the four processes.

$$TEC = x1 * TEC_{m,i1} + x2 * TEC_{m,i2} + x3 * TEC_{m,i3} + x4 * TEC_{m,i4} \tag{12}$$

Finally, the total amount of energy that the mobile phone has available to run the process until the device is charged will be obtained. Significant amounts of energy are spent on operations like operating systems, screen display, and phone calls. It is not possible to measure these energy values separately, and they do not change much over time. It will be more accurate to consider the energy that a mobile phone has just to run the process. By taking advantage of past process records, the average completion times and average power values of the processes run on the two specified time intervals will be obtained. As a result of multiplying these two values, the amount of energy the mobile device has to run the process until it is charged will be estimated.

To summarize this section, in the first step, the mobile phone's next charging time is estimated. Then, between the instantaneous time and the charging time, the processes that the user might run were determined and the total energy cost required for these processes was calculated. Finally, the

amount of energy that the phone has available to run the process at the two specified time intervals is estimated. As a result, information about the battery level is obtained by comparing the amount of energy needed for the processes to be run with the amount of energy the phone has.

4.3. Decision Algorithm

In the previous chapters, if the mobile phone's next charging time can be estimated, the energy cost required for its possible processes between the decision time and the charging time is expressed as E_M . In these two time periods, the amount of energy that the mobile phone has to run the processes is determined as E_S . In addition, the state of the mobile phone battery plugged in is expressed as B_{state} .

The following information is obtained for each p process for use in the algorithm:

- $T_{m,p}$ = Estimated completion time of p process in the mobile environment
- $T_{c,p}$ = Estimated completion time of p process in the cloud environment
- $TEC_{m,p}$ = Energy cost estimation if the p process is run in the mobile environment
- $TEC_{c,p}$ = Energy cost estimation if the p process is run in the cloud environment

The weight ratios determined for critical battery level are shown in Table 6. The weight ratios of normal battery level are shown in Table 7. Weight ratios were used in similar studies [11, 37, 38]

Table 6. Weight ratios for critical battery level

Criterion	Symbol	Weight ratio
Energy	E_Y	0,8
Performance	P_Y	0,2

Table 7. Weight ratios for normal battery level

Criterion	Symbol	Weight ratio
Energy	E_N	0,8
Performance	P_N	0,2

The pseudo code of the decision algorithm is shown in Fig. 4. In the first phase of the algorithm, the costs of running a process in the mobile and cloud environments are calculated. At this stage, E_M , E_S and B_{state} values are taken into consideration. In other words, if the mobile phone has sufficient energy to run the required processes until the next charging time, or if the mobile phone battery is plugged in, the weight values for the normal battery level will be used during the calculation of mobile and cloud cost values. Otherwise, the weight values for the critical battery level will be used. Thus, an energy priority decision can be made. The cost values for the mobile and cloud environments are obtained as a result of multiplying the value of proportions of the energy and time values with the determined weight ratios. Later, the costs of the mobile and cloud environments are compared. If the cloud environment cost is lower than the cost of the mobile environment, the relevant process is transferred to the cloud; otherwise, the process is run in the mobile environment.

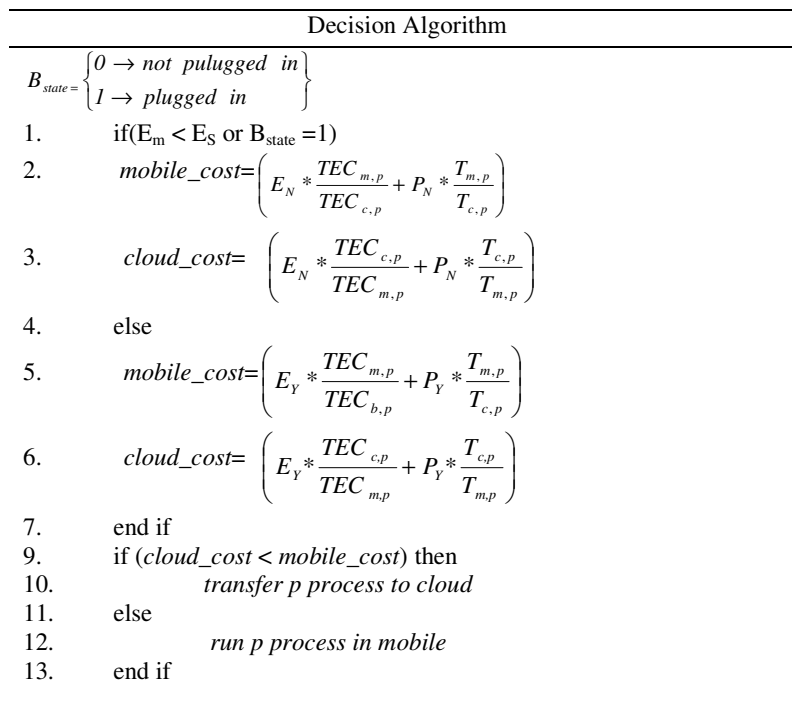


Figure 4. Pseudo code of decision algorithm.

5. EXPERIMENTAL RESULTS

The developed application has been used in the real environment by 7 mobile users. The experimental study comprises two phases. In the first stage, a mobile application that could not transfer its processes to the cloud was used for 15 days. In the second phase, a mobile cloud application that could transfer its processes to the cloud was used for 15 days. At the end of the experimental study, the mobile application data was transferred through web services to the central database in the cloud environment.

While the mobile device users were identified for the experimental study, users were considered to have different context information and had mobile devices with different specifications. The brands and models of users' mobile devices are listed in Table 8.

Table 8. Mobile phone models of users

User	Mobile phone model
User 1	Samsung Galaxy A8
User 2	General Mobile Discovery 2
User 3	Samsung Note 4
User 4	Samsung Galaxy S6
User 5	LG G4
User 6	General Mobile 5 plus
User 7	LG G4

The mobile application's ".apk" extension installation file is available on the internet to mobile phone users. The mobile application installed on the specified mobile phones ran on the backplane without user interaction.

5.1. Comparison of Mobile Application and Mobile Cloud Applications

The application used in the first phase of the experimental study is called "mobile application" and the application used in the second phase and which was capable of transferring processes to the cloud is called "mobile cloud application".

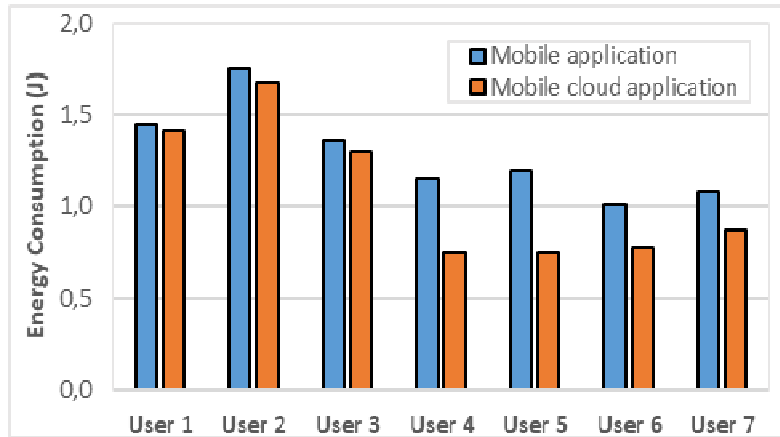


Figure 5. Comparison of mobile application and mobile cloud applications in terms of energy consumption

In Fig. 5, the mobile and mobile cloud applications are compared for each user in terms of average energy consumption. In Fig. 6, the mobile and mobile cloud applications are compared in terms of the completion times of the processes. If a user-based assessment is performed by looking at Fig. 5 and Fig. 6, it can be said that the mobile cloud application provides better performance for all users in terms of energy consumption and process completion times. It can also be said that these performance values are higher for user 4 and user 5 and lower for user 1, user 2 and user 3.

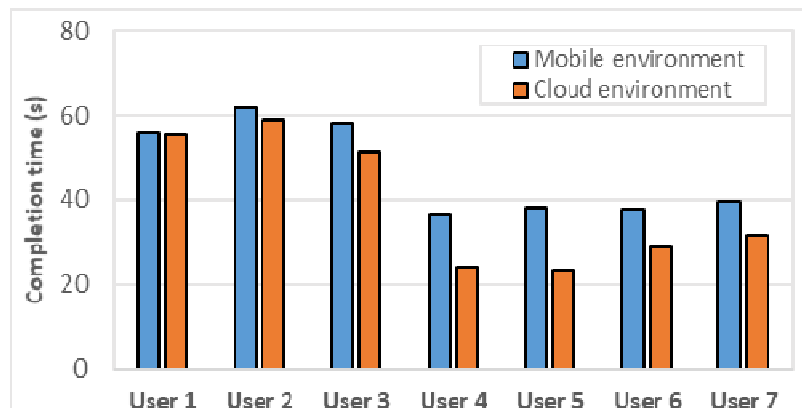


Figure 6. Comparison of mobile application and mobile cloud applications in terms of completion time.

5.2. Comparison of Mobile Environment and Mobile Cloud Environments

In this section, the data from the mobile cloud application used in the second 15 days of the experimental study was compared in terms of the mobile and cloud environments.

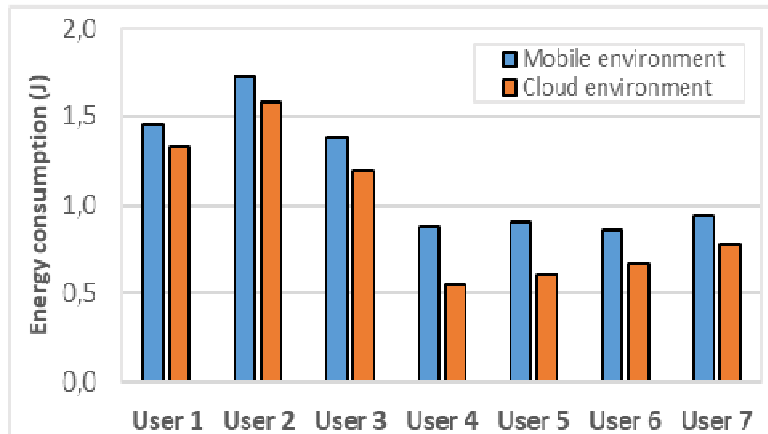


Fig. 7. Comparison of mobile and cloud environments in terms of energy consumption.

Fig. 7 shows the average energy consumption values spent in the mobile and cloud environments for each user. This figure shows that the average energy spent in the cloud environment is less than in the mobile environment for each user. For users 4 and 5, it can be said that about 50% of energy saving is achieved by transferring processes to the cloud.

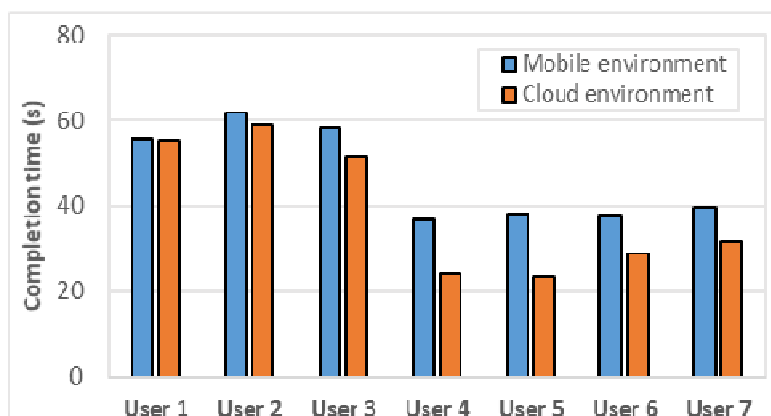


Figure 8. Comparison of mobile and cloud environments in terms of completion time.

In Fig. 8, average completion times within the mobile and cloud environments are compared for each user. This figure shows that the process completion times for users 1, 2 and 3 are longer than for other users. This may be because the technical capacities of these users' mobile devices are more limited than the others'.

5.3. Performance Evaluation Based On Internet Connection Type

Wi-Fi, EDGE, 3G and 4.5G (LTE) internet connection types were used when transferring processes to the cloud. Fig. 9 shows the number of processes transferred to the cloud by these connection types for each user. When the figure is examined, it is seen that Wi-Fi is the most used connection type in the cloud transfer stages for all users. After Wi-Fi, the most used connection type was 4.5G.

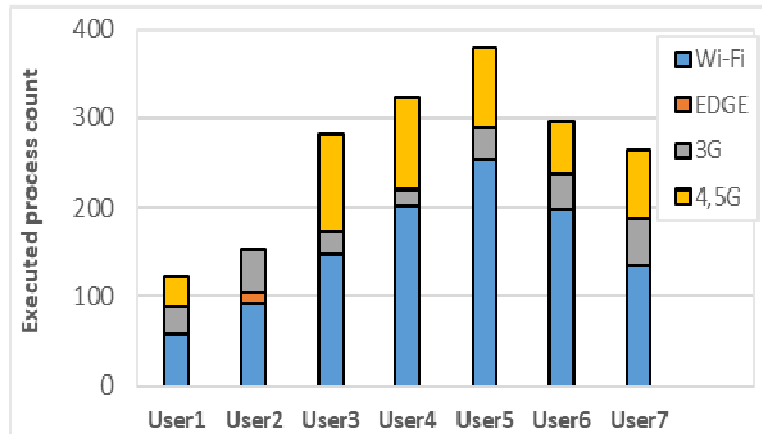


Figure 9. Number of processes running by connection types.

Fig. 10 shows average energy consumption values by each connection type for each user and Fig. 11 shows their average completion times. When the figures are examined, it can be said that the processes that are generally operated with a Wi-Fi connection are completed in less time using less energy. After Wi-Fi, the 4.5G connection type provides the best performance in terms of energy consumption and completion time. It can even be said that the 4.5G connection type performs better than Wi-Fi for user 5 and user 7. The reason for this is that these users have the appropriate 4.5G connection in their location.

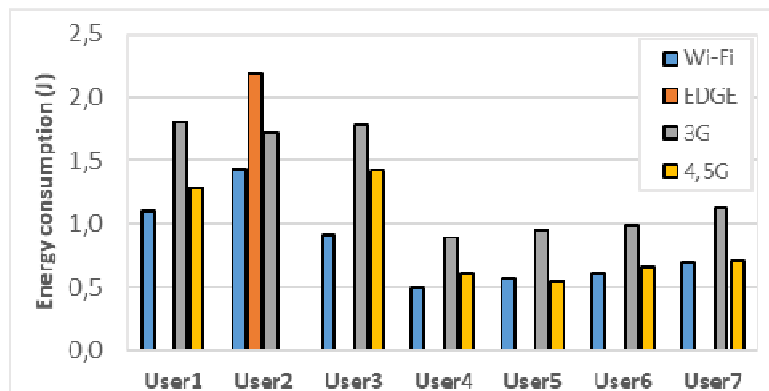


Figure 10. Comparison of connection types in terms of energy consumption.

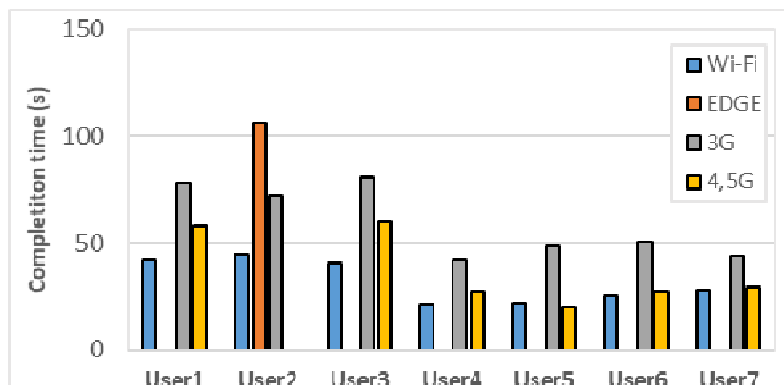


Figure 11. Comparison of connection types in terms of completion types.

5.4. Comparing Mobile and Cloud Environments Based On Process Type

When the graphs of energy and time values are examined, it is seen that, when compared to the other processes, the eight queens puzzle process is completed in less time using less energy, and the face-detection process is completed in more time using more energy. The reason that the eight queens puzzle process can be completed in a shorter period using less energy in the mobile environment is that this process requires fewer mobile device resources than do other processes. This process can be run with lower values in the cloud environment because the size of the file needing to be transferred to the cloud for this process is smaller than in other processes. The fact that the face-detection process is completed in the mobile environment over a longer period using more energy than other processes may indicate that this process uses more mobile resources than do other processes. The reason this process is completed in the cloud environment using more values than other processes is that the size of the file needing to be transferred to the cloud is larger than in other processes.

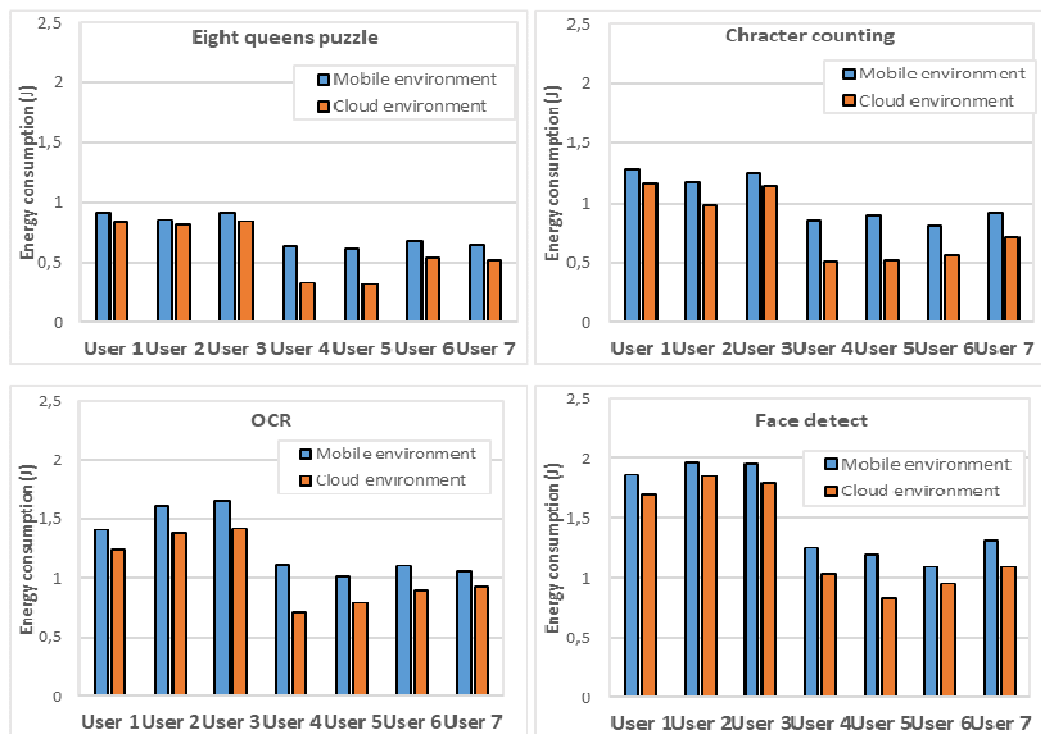


Figure 12. Comparison of process types in terms of energy consumption.

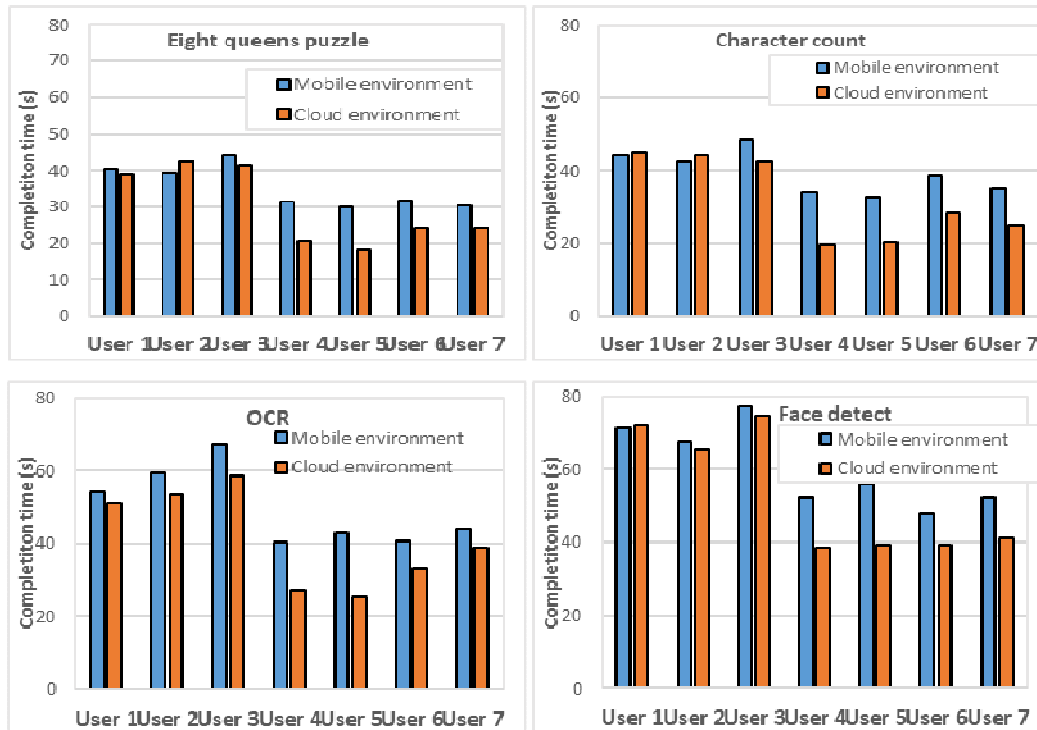


Figure 13. Comparison of process types in terms of completion time

6. CONCLUSION

In order to be able to make energy priority calculations at the decision stages, the next charging time of the mobile device is estimated and it is determined whether the current battery level is sufficient for normal use until the next charge. Based on this information, the weight ratios in the decision algorithm are determined. When the estimates on mobile device charging times were examined, it was determined that accurate estimates are made for users who regularly charge their phones. For users without a regular charging habit, many prediction errors are made early on, but these errors gradually decrease over time.

Comparing the results obtained by the mobile applications used in the first phase and mobile cloud applications used in the second phase, the mobile cloud application showed better performance at varying rates for each user in terms of completion time and energy consumption. When the mobile and cloud environment performances are evaluated for each user, it was determined that in the cloud environment, less energy usage and shorter run times are achieved for each user. Due to the cloud environment, some users have achieved an approximately 50% better performance in terms of energy and performance.

According to the results, it can be interpreted that the processes needing more mobile device resources consume more energy in the mobile environment and take longer to complete, whereas in the cloud environment, the size of the file transferred to the cloud is the decisive factor in energy usage and completion time.

The most important result of this study is the conclusion that it may be possible to take advantage of mobile cloud computing opportunities under appropriate conditions. If mobile application developers and companies offer the opportunity to transfer specific processes to the cloud for

their applications, it will be possible for their applications to perform well without using much energy.

REFERENCES

- [1] S. Hossain, (2014) “What is mobile cloud computing?”, <https://www.ibm.com/blogs/cloud-computing/2014/01/02/what-is-mobile-cloud-computing>.
- [2] T.Y. Lin, T.A. Lin, C.H. Hsu and C.T. King, (2013) “Context-aware decision engine for mobile cloud offloading”, IEEE Wireless Communications and Networking Conference Workshops (WCNCW), Shanghai, pp. 111–116.
- [3] H. Wu, Q. Wang and K. Wolter, (2013) “Tradeoff between performance improvement and energy saving in mobile cloud offloading systems”, IEEE International Conference on Communications Workshops (ICC), Berlin, pp728–732.
- [4] F. Xia, F. Ding, J. Li, X. Kong, L. T. Yang and J. Ma, (2014) “Phone2Cloud: Exploiting computation offloading for energy saving on smartphones in mobile cloud computing”, Information Systems Frontiers, Vol. 16, pp95–111.
- [5] Y. Hao, M. Chen, L. Hu, M S. Hossain and A. Ghoneim, (2018) “Energy efficient task caching and offloading for mobile edge computing”, IEEE Access, Vol. 6, pp11365–11373.
- [6] S. Kosta, A. Aucinas, P. Hui, R. Mortier and X. Zhang, (2012) “ThinkAir: Dynamic resource allocation and parallel execution in the cloud for mobile code offloading”, INFOCOM 2012 Proceedings IEEE, Orlando, pp945–953.
- [7] B.G. Chun, S. Ihm, P. Maniatis, M. Naik and A. Patti, (2011) “CloneCloud: Elastic execution between mobile device and cloud”, Proceedings of the Sixth Conference on Computer Systems, New York, pp301–314.
- [8] W. T. Su and K. S. Ng, (2013) “Mobile cloud with smart offloading system”, IEEE/CIC International Conference on Communications in China (ICCC), Xi'an, pp680–685.
- [9] AsyncTask, (2016), <https://developer.android.com/reference/android/os/AsyncTask.html>.
- [10] K. Kumar, J. Liu, Y. H. Lu and B. Bhargava, (2013) “A survey of computation offloading for mobile systems”, Mob. Netw. Appl., Vol. 18, pp129–140.
- [11] B. Zhou, A. V. Dastjerdi, R. N. Calheiros, S. N. Srirama and R. Buyya, (2015) “A context sensitive offloading scheme for mobile cloud computing service”, IEEE 8th International Conference on Cloud Computing, New York, pp869–876.
- [12] C. M. Magurawalage, K. Yang, L. Hu and J. Zhang, (2014) “Energy-efficient and network-aware offloading algorithm for mobile cloud computing”, Comput. Netw., Vol. 74, pp22–33.
- [13] K. H. Lim and B. D. Lee, (2014) “History-based dynamic estimation of energy consumption for mobile applications”, 16th International Conference on Advanced Communication Technology, Pyeongchang, pp714–718.
- [14] R. Aldmour, S. Yousef, M. Yaghi, S. Tapaswi, K. Pattanaik and M. Cole, (2017) “New cloud offloading algorithm for better energy consumption and process time”, Int. J. Syst. Assur. Eng. Manag., Vol. 8, pp730–733.
- [15] L. Li, X. Zhang, K. Liu, F. Jiang and J. Peng, (2018) “An energy-aware task offloading mechanism in multiuser mobile-edge cloud computing”, Mobile Information Systems.
- [16] M. M. Islam, M. A. Razzaque, M. M. Hassan, W. N. Ismail and B. Song, (2017) “Mobile cloud-based big healthcare data processing in smart cities”, IEEE Access, Vol. 5, pp11887–11899.
- [17] S. M. A. Karim and J. J. Prevost, (2017) “A machine learning based approach to mobile cloud offloading”, Computing Conference, London, pp675–67.
- [18] M. Jia, J. Cao and L. Yang, (2014) “Heuristic offloading of concurrent tasks for computation-intensive applications in mobile cloud computing”, IEEE Conference on Computer Communications Workshops, Toronto, pp352–357.
- [19] H. Eom, P. S. Juste, R. Figueiredo, O. Tickoo, R. Illikkal and R. Iyer, (2013) “Machine learning-based runtime scheduler for mobile offloading framework”, IEEE/ACM 6th International Conference on Utility and Cloud Computing, Dresden, pp17–25.
- [20] H. Flores and S. Srirama, (2013) “Adaptive code offloading for mobile cloud applications: exploiting fuzzy sets and evidence-based learning”, Proceeding of the Fourth ACM Workshop on Mobile Cloud Computing and Services, New York, pp9–16.
- [21] M. R. Ra, (2013) “Cloud-enabled mobile sensing systems”, PhD Thesis, University of Southern California.

- [22] Gaurav, N. Kaushik and J. Bhardwaj, (2014) “A computation offloading framework to optimize makespan in mobile cloud computing environment”, *Int. J. Adv. Comput. Res.*, Vol. 4, pp442-449.
- [23] R. R. de Oliveira, N. M. S. Schirmer, M. Machry and T. C. Ferreto, (2017) “A transparent code offloading technique for Android devices”, 13th International Wireless Communications and Mobile Computing Conference, Valencia, pp1078–1083.
- [24] S. Yang et al., (2013) “Fast dynamic execution offloading for efficient mobile cloud computing”, *IEEE International Conference on Pervasive Computing and Communications*, San Diego, pp 20–28.
- [25] E. Meskar, T. D. Todd, D. Zhao and G. Karakostas, (2017) “Energy aware offloading for competing users on a shared communication channel”, *IEEE Transactions on Mobile Computing*, Vol. 16, pp87-96.
- [26] S. Yang, D. Kwon, H. Yi, Y. Cho, Y. Kwon and Y. Paek, (2014) “Techniques to minimize state transfer costs for dynamic execution offloading in mobile cloud computing”, *IEEE Transactions On Mobile Computing*, Vol. 13, pp2648-2660.
- [27] SQLite, (2016) <http://www.sqlite.org>.
- [28] M. E. Khoda, M. A. Razzaque, A. Almogren, M. M. Hassan, A. Alamri and A. Alelaiwi, (2016) “Efficient computation offloading decision in mobile cloud computing over 5G network”, *Mob. Netw. Appl.*, Vol. 21, pp777–792.
- [29] M. R. Rahimi, N. Venkatasubramanian and A. V. Vasilakos, (2013) “MuSIC: Mobility-aware optimal service allocation in mobile cloud computing”, *IEEE Sixth International Conference on Cloud Computing*, Santa Clara, pp75–82.
- [30] K. Pandi and H. Charaf., (2013) “Performance metrics based mobile resource management”, *IEEE International Conference on System Science and Engineering*, Budapest, pp329-333.
- [31] A. R. S. Nugroho, (2016) “Exploring privacy leakage from the resource usage patterns of mobile apps”, PhD Thesis, University Of Arkansas.
- [32] E. Cuervo et al., (2010) “MAUI: Making smartphones last longer with code offload”, *Proceedings of the 8th International Conference on Mobile Systems, Applications, and Services*, New York, pp 49–62.
- [33] C. Chang, (2015) “A Framework for Energy-efficient Mobile Cloud Offloading”, University of Tartu.
- [34] N. I. M. Enzai and M. Tang, (2014) “A taxonomy of computation offloading in mobile cloud computing”, *2nd IEEE International Conference on Mobile Cloud Computing, Services, and Engineering*, Oxford, pp19–28.
- [35] K. Korhonen, (2011) “Predicting mobile devices battery life”, Master Thesis, Aalto University Finland.
- [36] J. M. Kang, C. P. Park, S. S. Seo, M. J. Choi, J. W. Hong (2008) “User-Centric predicting for battery lifetime of mobile devices”, *Challenges for Next Generation Network Operations and Service Management*, pp531-534.
- [37] L. Pu, J. Xu, X. Jin and J. Zhang, (2013) “SmartVirtCloud: virtual cloud assisted application offloading execution at mobile devices' discretion”, *Wireless Communications and Networking Conference*, Shanghai, pp4398–4403.
- [38] Y. Tao, Y. Zhang, and Y. Ji, (2015) “Efficient computation offloading strategies for mobile cloud computing”, *IEEE 29th International Conference on Advanced Information Networking and Applications*, Gwangju, pp626–633.
- [39] G. Orsini, D. Bade and W. Lamersdorf, (2018) “CloudAware: Empowering context-aware self-adaptation for mobile applications”, *Transactions on Emerging Telecommunications Technologies*, Vol. 29.
- [40] S. Yan, C. Shanzhi and X. Xiang, (2018) “MAGA: A mobility-aware computation offloading decision for distributed mobile cloud computing”, *IEEE Internet of Things Journal*, Vol. 5, pp164-174.

AUTHORS

Dr. Mustafa Tanrıverdi received the Ph.D. degree in Management Information System from Gazi University, Ankara, Turkey, in 2017. He was working in Department of Computer in Gazi University, Turkey until 2007. He has research interest are mobile applications, cloud computing, software development and blockchain.



Dr. M. Ali Akcayol is professor in Engineering Faculty at Gazi University, Turkey. His research interests are mobile wireless networks, artificial intelligence, cloud computing, web technologies, web mining, smart buildings. He has published a lot of research paper in national, international journals and conferences in her areas of expertise.

