IINTERNATIONAL CONFERENCE PROCEEDINGS 00004 (2022) | DOI : https://doi.org/10.29103/micoms.v3i.42
E-ISSN: 2963-2536

Optimizing Multi-Time Notifications Using Q-Learning

Muhammad Fikry[1,3,*], Muhammad Iqbal[2,3]

[1] *Graduate School of Life Science and Systems Engineering, Kyushu Institute of Technology, Japan*
[2] *Graduate School of Human-Environment Studies, Kyushu University, Japan*
[3] *Informatics, Universitas Malikussaleh, Indonesia*
[*] *Corresponding author. Email: fikry.muhammad622@mail.kyutech.jp*

ABSTRACT
In this research, we propose time optimization for notifications to assist users in remembering their actions by taking into consideration the amount of time it takes for them to respond and react. Using the Q-Learning algorithm, this proposal calculates when the best time to send notifications to users' smartphones in order to remind them of something important. The time at which the message is sent will be adjusted depending on the replies of prior users, which may be transformed into feedback at any time that is convenient. Notifications will be sent out, either repetitively or not, depending on the appropriate time for each individual, with the goal of ensuring that users do not forget about activities that they have planned. The results of testing our technique using the dataset show that it may be used to improve the time at which notifications are issued to recipients. It is possible to experiment with a variety of different times for the delivery of alerts in order to determine which of these periods is most successful for prompting users to take action. As a consequence of this, the algorithm is able to accommodate specific characteristics of individuals and find solutions to problems using a variety of standard operating procedures. Our proposal has the potential to successfully maintain the notification execution time at the intended level, which will prevent users from becoming concerned about the volume of notifications. Users who do not see the notification initially have the opportunity to do so at a later time step, which guarantees that activity data will still be collected.

Keywords: *Notification, Smartphone, Reminder System, Q-Learning.*

1. INTRODUCTION

The act of remembering to carry out a predetermined action or remembering a predetermined purpose at some point in the future is an example of the kind of memory known as prospective memory. Tasks that need prospective memory are widespread in day-to-day life and vary from very easy life-or-death scenarios to severe cases. Anyone can experience prospective memory, so it is necessary to design assistive technology that can remember activities based on existing schedules. The memory that is stored in the brain may be broken down into two categories: short term and long term. Information that is maintained in the brain for a short period of time but will be lost if the person does not recall it or does not use it often, such as memorizing the name or telephone number of someone you just met, is an example of short-term memory. On the other hand, if you practice something often, such as how to operate a car or make a meal, your short-term memory may convert into long-term memory. Memory is something that originates in the human brain, which, due to the natural process of aging, may lose some of its capacity, which in turn can have an effect on memory. This is a result of the numerous human activities that may cause people to miss scheduled events. People who forget they have prior obligations may need to postpone their plans. Even those who lead routine lives occasionally forget appointments. A reminder system can assist everyone in keeping track of critical information and helping to recall important people. Some reminder systems merely send notifications based on an event's occurrence at a predefined time. People can personalize their own reminder system to suit their preferences. They can generate one, two, or more notifications and specify the reminder time to occur on schedule, a few hours before schedule, or only minutes before schedule. However, busy work can prevent them from noticing these reminders, and for those with memory impairment, forgetting their phones can prevent them from seeing the reminder or engaging in certain activities can cause them to forget despite having already been reminded by the system. For these people, one or two notifications may not be sufficient, while an excessive number of notifications may become annoying.

To overcome this problem, we aim to find the best time to send notifications using the Q-Learning algorithm in order to remind the user before the schedule at the right time with or without repetition that does not distract the user from the notification. Q-Learning is one of the reinforcement learning algorithms that learns based on the agent's experience; memory technology utilizing this method is seen useful for overcoming difficulties in establishing fixed rules for each individual due to their diverse conditions and actions. Based on the user's reaction history, we optimize the time and reduce the amount of delivered alerts using the q-learning algorithm. so that the user will receive timely feedback to inform them of the upcoming task. Depending on the response, different people will receive notices and reminders at different times.

2. RELATED WORK

In most cases, the user is required to have prior knowledge of the schedule time in order to successfully create a reminder for himself. It does not always work because if the user is in an unknown condition and he wants to be reminded, it will be impossible for the user to set a reminder in this circumstance. This is one of the reasons why it does not always work. In addition, based reminders are created whenever the user is getting close to the time that was scheduled. If the user receives a notification from the reminder application at an inappropriate moment, such as when driving, this may distract the user while they are driving, which may increase the risk of an accident. Because of this problem, app developers were pushed to reduce the amount of work required by consolidating this functionality into a single app, which in turn improved the way in which consumers used reminders. People rely on a variety of methods to assist them with the task of remembering things in their day-to-day lives, such as calendars [1], diaries [2], post-it notes, and smartphone [3][4].

Many studies relating to reminder systems have been carried out with various technologies to help humans because they have benefits in various aspects of life, such as studying, appointments, or health checks. The reason for this is because reminder systems have been shown to help humans in a variety of situations. People who have cognitive problems that are becoming worse may benefit from reminder, such as instructional prompts and scheduling assistance [5]. Users get information on their desktop computers [5], laptops, and smartphone [6][7][8][9][10] in a variety of ways, one of which is via notification. Notifications may be sent to consumers in the form of visual cues, audio signals, or haptic alerts [11]. They are created by applications in order to communicate important information to users. On the other hand, there is a growing number of alerts that demand the attention of smartphone users, and these notifications often occur at unsuitable times or include irrelevant material. Previous study has shown that alerts at unsuitable times may raise anxiety and interfere with the time it takes to complete a job. Notifications can induce distraction if they are received at the wrong times. People have a hard time getting back on track after being side tracked by anything like a phone call or an instant message, and this difficulty is compounded when the original work was one that required a high level of mental effort since this makes the impact even more obvious [12].

Because notifications on the smartphone can originate from a variety of applications with the intention of conveying information to the user, while the user may have certain priorities regarding events or actions that must be carried out, time plays an important role in the reminder user's responsibility to check the time. One of the algorithms that may be utilized is called q-learning, and its purpose is to optimize the amount of time spent delivering alerts. Q-learning is a method of reinforcement learning that is used to determine the best action-selection strategy [13] by making use of a quality function that is referred to as the Q function. Instead of identifying the potential value of the state to which one is relocating, the concept behind Q-Learning is to evaluate the efficacy of the activity that was made in order to move to that state. Using the newly revised Q-table, we are now able to bring the Q-values for starting at the beginning and progressing to the right up to date. This method is used for optimization by a number of scholars, such as Xin Hu et al [14] employs Q-learning to assess the performance in terms of cumulative profits by maximizing various types of value functions for the purpose of optimizing portfolios with asset allocation between risky and risk-free products, Sayed Amir Hoseini et al [15] modeled trajectory optimization using the Q-Learning method, which can outperform two benchmark techniques known as random route planning and static drift, Mukai et al [16] via the use of Q-learning, optimize routes for on-demand bus systems.

In the course of our work, we will optimize multi-time notifications; we will provide a lot of time to send notifications; and we will find the best time of the lot so that the information conveyed can be exactly when the user needs it. All of these things will help ensure that the user receives the information exactly when they require it.

3. MODELING AND OPTIMIZATION

This section outlines our process for deciding when to notify users of new content. Users can specify the scheduled time in the system that we suggest in order to meet their demands, but the system will deliver notifications based on the time that has been optimized rather than the time that the user has selected. The system will decide whether to apply a notify or silent action to the user. Users are requested to respond by selecting the alternatives that appear in the notice, such as accept or dismiss, when the system sends notifications to them. Silent means the system does not send notifications, while notify means the system will send notifications to the user, the user is asked to respond by selecting the options that appear in the notification, such as accepting or rejecting. The system will assume that the user ignores the message at that time if they do not choose that choice because they forgot or for some other reason. We present a general overview of the system operations in Figure 1.

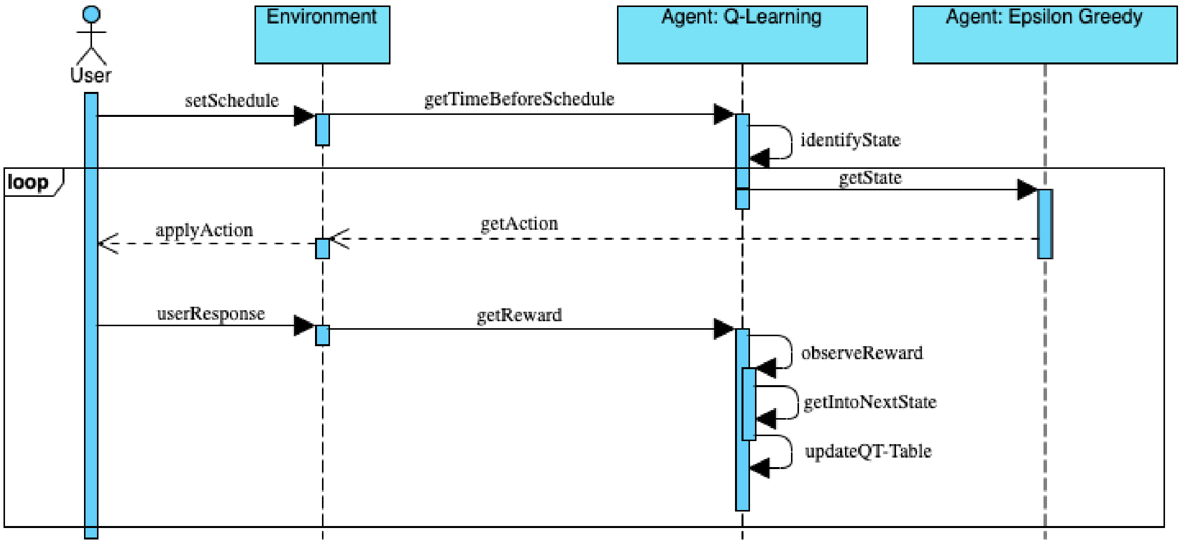

Figure 1 An overview of the proposed method

An overview of the proposed method, the agent is a system, while the environment is the user's smartphone. The agent looks at the user's context and decides whether to deliver notifications or not by doing nothing. The message on the notification that appears gives the user the option to respond, and the agent will use that response to determine the reward.

The interaction between the agent and environment consists of the agent making observations to get a representation of the environment, known as the state, and then acting in accordance with its policies. The environment transitions from its present condition to the next one and delivers a reward based on the action made. The agent will optimize the number of future reward discounts accumulated. We employ Q-Learning as a reinforcement learning algorithm. Initially, the q-learning algorithm we use initializes for all states at time t and actions at time t.

$$S_t \in S; A_t \in A; \tag{1}$$

where t is the step time to send the notification if the action is notify; state is the product of time and input; input is the possible response from user; action is notify or not. We assign 0 to the initial value for t.

$$t = 0 \tag{2}$$

The system will start with S_0, At the time step t, the system choose action A_t from the maxima and the greedy epsilon method is applied to this part.

$$A_t = arg\,max_{a\in A}\, Q(S_t, a) \tag{3}$$

The system apply the action A_t, observe the reward (Eq. (4)) and go tothe next state (Eq. (5)) based on the user's response.

$$R_{t+1}(R_{t+1} = r(S_t, A_t)) \quad (4)$$

$$S_{t+1}\,(S_{t+1} = \delta(S_t, A_t)) \quad (5)$$

Reward is a function of action A at time t in state S at time t, whereas the next state is decided by the transition state S at time t after executing action A at time t. The agent will then compute a q-value similar to Eq. (6) in order to update the q-table.

$$\left(S_t, A_t\right) \leftarrow Q\left(S_t, A_t\right) + \alpha(R_{t+1} + \gamma\, max_{a\in A} Q\left(S_{t+1}, a\right) - Q(S_t, A_t)) \quad (6)$$

Because there is only one possible response from the user at each time in our method, and that response can be either accept, dismiss, or ignore, which means that each time has three states, the number of states that we have is equal to the number of alternative times multiplied by the number of possible responses.

4. EXPERIMENTAL SETUP

We conducted an experiment with artificially generated data in order to evaluate our proposed model. In this experiment, we used variables such as ActivityID, UserID, TimeSchedule, Action, TimeAction, TimeResponse, and UserResponse. There were a total of 124 notify actions and 126 silent actions. We created a synthetic dataset by giving random user responses. When we send notifications, 30.65% of the time the response is accept, 31.45% of the time the response is dismiss. At this point in the process, the system will solicit user answers that are made immediately accessible, agents will learn via trial and error while interacting with the environment, and they will get incentives for their activities. From this synthesis data, we continue to shape the data created directly by the system in order to mimic learning agents directly from their surroundings. As a result of the tests that we carried out, we obtained 111 notification actions and 139 quiet actions. When deciding on a course of action, the agent will sometimes engage in random exploration with a probability of and instead pick the best course of action most of the time with a probability of $1-\epsilon$.

In order to assess the efficacy of the approach that we have presented, a first experiment has been carried out to monitor the changes in status. Each instance has three states, and the transition between states may be understood if one assumes that the user will always accept, reject, or ignore the action, and that the action will always be quiet. We carry out synthetic simulations using the following performance metrics, which have been described in this paper: the number of warnings, the user response rate, and the time optimization. The number of notifications is the total number of alerts that are sent to users in order to remind them of the action that they are about to do. Time optimization is connected to the current state of the Q-table and serves as an essential criterion for determining the next course of action, which may be to send alerts or to remain quiet. It is referred to as the proportion of items inside the Q-table of our suggested algorithm that are updated.

5. RESULT

As a result of the algorithm's ability to learn a greater response rate over time, we have a number of different options available to us when deciding when to inform the user. Because each time has three alternative replies, which results in three states for each time, we perform tests to examine the transition between states by assuming the user's reaction is always accept, dismiss, or ignore, and the action is always quiet. This allows us to see how the states change. According to the findings, transitions between states might take place not from one state to the next at the same time but rather from the present state to the next time in the future.

The subsequent experiment that we carried out was one in which we utilized data that had been artificially generated. Synthetically, our data is generated manually, and then it is entered into the system. The system accepts the input, and it calculates it to get a q-table. Then, we do the process again using the q-learning algorithm by applying random user responses, which generates new data with an average response rate of 62%. In Figure 2, we present the total number of notifications that were received. The agent in the second process learns more directly from the environment, in contrast to the agent in the first process, who does not learn directly from the environment. This is the primary difference between these two processes.

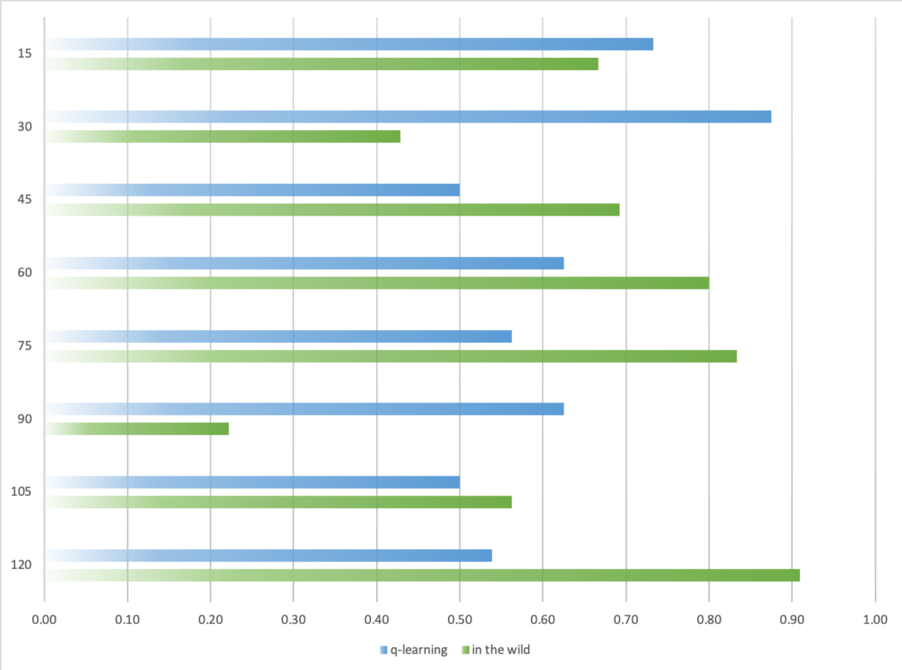

Figure 2 The number of notifications

As a consequence of this, the method accepts the user response that is processed using our proposed algorithm better. This is due to the fact that the proposal is able to send notifications at the appropriate time. The appropriate time for each user is determined by whether they prefer the time to be close to or far from the time they set. Figure 3 displays the findings that were obtained. There will be an inverse relationship between the high average accept and the dismissal rate (see Figure 4).

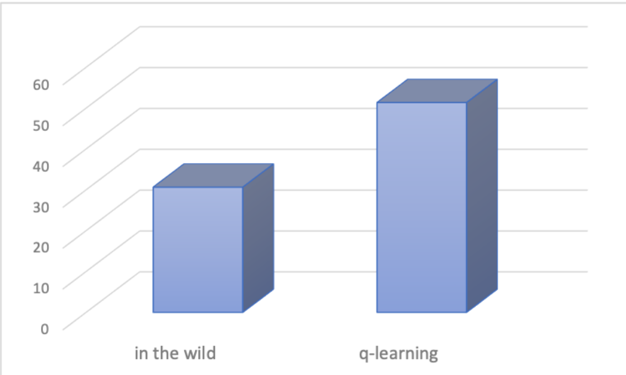

Figure 3 Accept rate

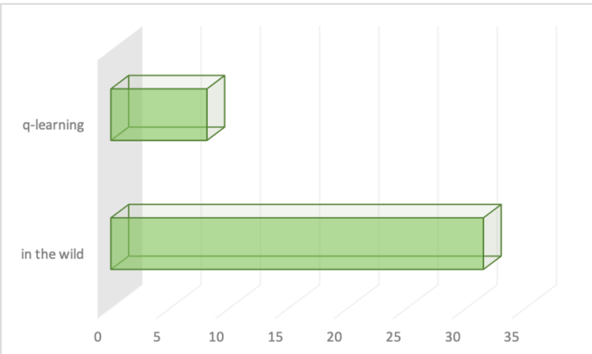

Figure 4 Dismiss Rate

Notification optimization is based on the response of each individual user, the reward return on the q-table is used as a guideline for the next action, and a comparison of reward returns is shown in Figure 5 and Figure 6. When we look at the reward return, the accept rate, and the dismiss rate, we can see that the number of notifications the user receives will increase proportionately with the frequency with which the user chooses to dismiss or ignore notifications. Since this is a dynamic process, the time at which notifications are sent can change again based on subsequent responses, including the number of notifications that are expected to be received. This indicates that the algorithm is able to accommodate variations in individual behavior, and the high accept rate demonstrates that the notification execution time is accurate.

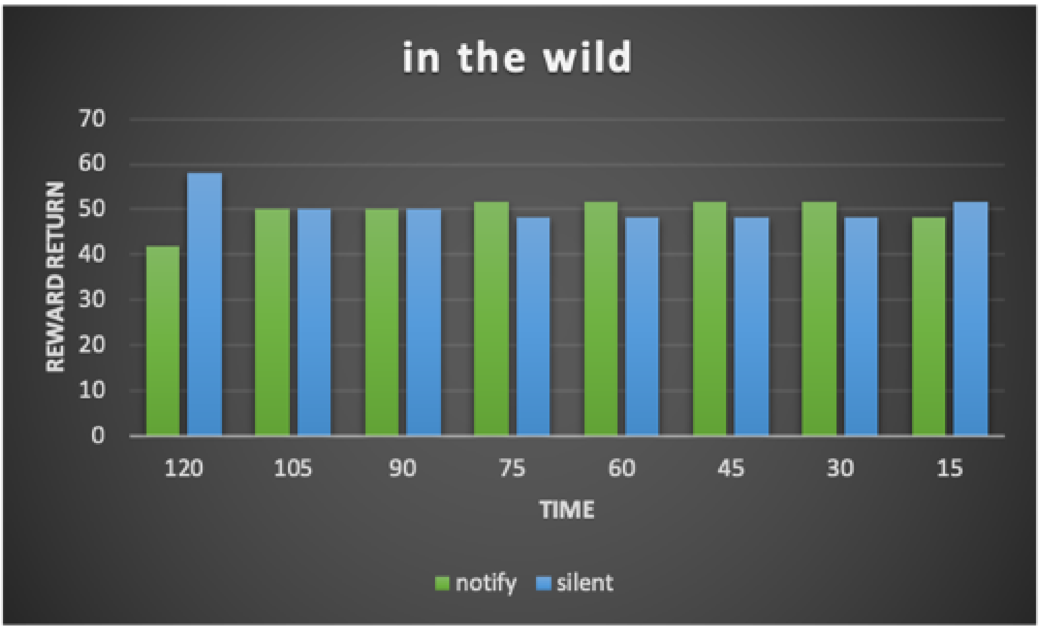

Figure 5 The average reward return in the wild

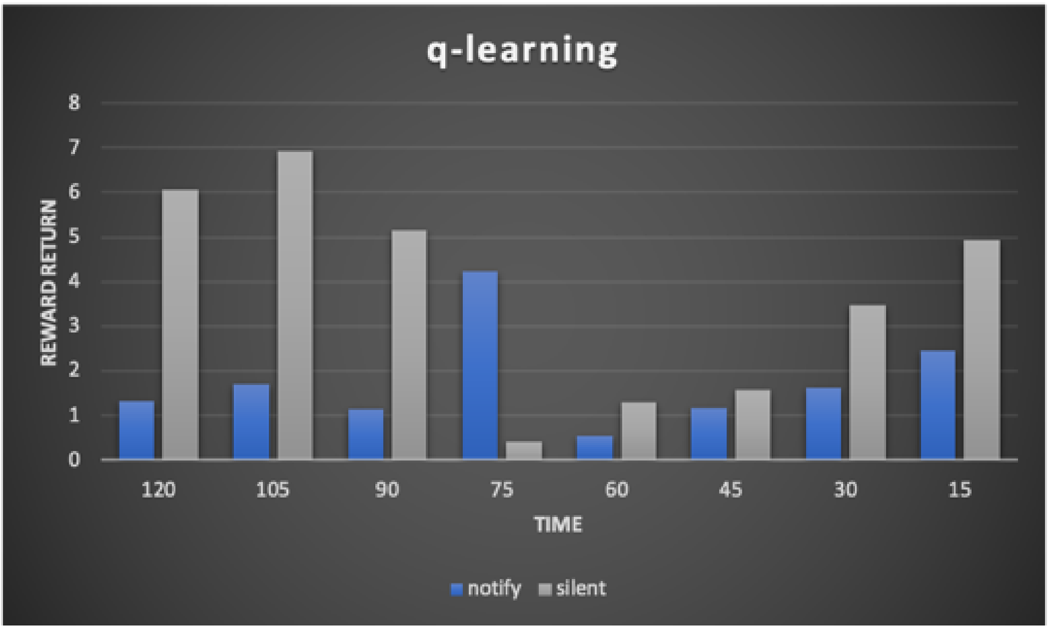

Figure 6 The average reward return by q-learning

The results of the experiments that have been carried out show that the model that we propose using the q-learning algorithm can optimize the time to send notifications. Additionally, the eight alternative times that are owned for sending notifications can be optimized in order to get the best time for each individual.

6. CONCLUSIONS

For the purpose of this paper, our strategy consists of determining the optimal number of notifications that should be sent to users. This is done so that people who have jobs that keep them extremely busy do not forget about the activities that they are going to do because one notification may not be enough for them. On the other hand, sending too many notifications can have the effect of being a nuisance. It also serves the purpose of overcoming the challenges that arise when attempting to establish permanent norms for each person due to the fact that people lead their lives in a variety of ways. In addition, the time in our proposal is dynamic, which means that if the user is able to appropriately reply to alerts, the system will optimize the time in order to reduce the total number of notifications that are delivered. The more trials that are run, the more the agent learns, and as a result, the agent is able to make more informed decisions on the selection of actions to send notifications or stay quiet, as well as the optimization of the time to send notifications.

We do not claim that this data can represent actual user data, so we cannot know how many notifications appear. Distraction from notifications can cause people to turn off notifications or cause people to ignore them frequently. This modeling allows us to effectively optimize our use of time, despite the fact that we do not claim that this data can represent actual user data. notification. We feel that in future work, investigating the quantity of alerts displayed at any one moment in the real world might offer better results for enhancing user engagement, and this is something that we will be pursuing.

AUTHORS' CONTRIBUTIONS

All authors contributed equally to the conceptualization, calculation, and analysis of data, as well as the result of the publication.

ACKNOWLEDGMENTS

This research is part of the ongoing doctoral program work. The author would like to thank Universitas Malikussaleh, the Asian Development Bank (ADB) and The Ministry of Research, Technology and Higher Education of the Republic of Indonesia for supporting my study through the Doctoral Degree Scholarship Program (Grant No. L3749-INO).

REFERENCES

[1] Andreassen, Maria, et al. "Feasibility of an intervention for patients with cognitive impairment using an interactive digital calendar with mobile phone reminders (RemindMe) to improve the performance of activities in everyday life." International journal of environmental research and public health 17.7 (2020): 2222.

[2] Zhang, Xiaoyi, Laura R. Pina, and James Fogarty. "Examining unlock journaling with diaries and reminders for in situ self-report in health and wellness." Proceedings of the 2016 CHI conference on human factors in computing systems. 2016.

[3] Ahmed, Q., Mujib, S.: Activity recognition using smartphone accelerometer and gyroscope sensors supporting context-based reminder systems. Context Aware Reminder System, Faculty of Computing at Blekinge Institute of Technology (2014)

[4] Khan, Nisar, and Fazlullah Khan. "Context based reminder system Supporting persons using Smartphone accelerometer data." (2013).

[5] Sahami Shirazi, Alireza, et al. "Large-scale assessment of mobile notifications." Proceedings of the SIGCHI conference on Human factors in computing systems. 2014.

[6] Pollack,M.E.,Brown,L.,Colbry,D.,McCarthy,C.E.,Orosz,C.,Peintner,B.,Ramakrishnan, S., Tsamardinos, I.: Autominder: An intelligent cognitive orthotic system for people with memory impairment. Robotics and autonomous systems 44(3-4), 273–282 (2003)

[7] Westermann, Tilo, Sebastian Möller, and Ina Wechsung. "Assessing the relationship between technical affinity, stress and notifications on smartphones." Proceedings of the 17th International Conference on Human-Computer Interaction with Mobile Devices and Services Adjunct. 2015.

[8] Kanjo, Eiman, Daria J. Kuss, and Chee Siang Ang. "NotiMind: utilizing responses to smart phone notifications as affective sensors." IEEE Access 5 (2017): 22023-22035.

[9] Weber, Dominik, Alireza Sahami Shirazi, and Niels Henze. "Towards smart notifications using research in the large." Proceedings of the 17th International Conference on Human-Computer Interaction with Mobile Devices and Services Adjunct. 2015.

[10] Gallud, Jose A., and Ricardo Tesoriero. "Smartphone notifications: A study on the sound to soundless tendency." Proceedings of the 17th International Conference on Human-Computer Interaction with Mobile Devices and Services Adjunct. 2015.

[11] Chen, H., Soh, Y.C.: A cooking assistance system for patients with alzheimers disease using reinforcement learning. International Journal of Information Technology 23(2) (2017)

[12] Leiva,L.,Böhmer,M.,Gehring,S.,Krüger,A.:Backtotheapp:thecostsofmobileapplica- tion interruptions. In: Proceedings of the 14th international conference on Human-computer interaction with mobile devices and services, pp. 291–294 (2012)

[13] Fikry, Muhammad, et al. "Analysis of Model-Free Reinforcement Learning Algorithm for Target Tracking." Co-ELITe: Journal of Computer Engineering, Electronics and Information Technology 1.1.

[14] Du, Xin, Jinjian Zhai, and Koupin Lv. "Algorithm trading using q-learning and recurrent reinforcement learning." positions 1.1 (2016).

[15] Hoseini, Sayed Amir, et al. "Trajectory optimization of flying energy sources using q-learning to recharge hotspot uavs." IEEE INFOCOM 2020-IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS). IEEE, 2020.

[16] Mukai, Naoto, Toyohide Watanabe, and Jun Feng. "Route optimization using Q-learning for on-demand bus systems." International Conference on Knowledge-Based and Intelligent Information and Engineering Systems. Springer, Berlin, Heidelberg, 2008.