Design and Implementation of Notification Information Survey System and Survey Results toward Use-side Adaptive Notification Management

Kenta Taki*, Yuki Matsuda*†, Yutaka Arakawa*‡, Keiichi Yasumoto*
* *Graduate school of Information Science, Nara Institute of Science and Technology*, Nara, Japan
Email: taki.kenta.th6@is.naist.jp, yukimat.jp@gmail.com, ara@is.naist.jp, yasumoto@is.naist.jp
† Research Fellow of Japan Society for the Promotion of Science
‡ JST Presto, Japan

***Abstract*—Lots of interrupt notification method have been studied, however, most of the existing research assumes that the applications do not control the notification timing except for the target application. However, if other applications are controlled by the same notification timing, concentration of interrupt timing will occur, and the effect of notification timing control may not be exerted. In addition, since the installed applications are different for each user, it is necessary to control notification timings taking into consideration the behaviors of all the applications installed on the user's smartphone. In this research, we define notification timing control considering behaviors of all installed applications as "Adaptive Notification Management", and conducted diversity surveys of notifications received by users. In this paper, we developed a system that acquires all notification information while excluding privacy. We report the experiment results actually collected using crowdsourcing, and discuss how to realize the application realizing adaptive notification management.**

***Index Terms*—Notification management, Mobile application, Adaptive computing, Context awareness, Interrupt notification, Mobile survey system.**

I. Introduction

Mobile notification is an important means for applications to actively provide information to users, however the number of information users receives increases year by year. However, notification at inappropriate timing will cause an increase in the user's stress and productivity, because there is a limit to the amount of information that can be perceived by humans. Therefore, many studies have been conducted on interruption at optimal timing using a wide range of variables such as context [1], environment [2], contents of message [3]. By controlling the timing based on these methods, it was possible to improve the response rate to the notification. However, when all notifications are controlled by not only a single application to be controlled but also a plurality of applications in the smartphone, One point concentration will occur against human attention.

In order to solve this problem, it is necessary to control by taking into consideration behaviors for all notifications of each user. However, the type of application installed on the user's smartphone and the status of permission for notifying the application is different for each user. For this reason, it is difficult for application developers who can not know the variety of application existing in the user's smartphone to take them into consideration for control. Even in the experimental system of Okoshi's research [1] which is actually carried out in collaboration with Yahoo, they are supposed to also be uncontrolled normal applications other than Yahoo's controlled applications. Therefore, if all the other applications notify the same breakpoint, the control performance may be adversely affected. In this research, we investigate the user's behavior against notifications of all applications existing on the user's smartphone, and then consider a method to neutrally personalize the notification timing at the use-side. This is called *adaptive notification management*.

In this research, in order to investigate the diversity of notifications received by users and the behaviors to each of them for realizing Adaptive Notification Management, we developed a system that can acquire all kinds of notification information while excluding privacy. We also used it to investigate general users who were recruited by crowdsourcing for four weeks and collected 95,910 notification data from 20 participants. We analyzed the collected data set statistically, organized the tasks in the adaptive notification management, and examined the possibility to realize individual optimization on the use-side.

II. Related work

A. Notification interrupt

The information notification from the mobile apps can be regarded as "interrupt" for the user. When this "interrupt" is inappropriate timing for the user, it causes an increase in user's stress and a decrease in productivity [4], [5].

B. Personal optimization for notifications

There is a study analyze the context of the notification on the use-side. Due to the popularization of context-aware computing and drastic improvement of the performance of mobile terminals, it became possible to provide services while tuning according to the behavior and environment of each user

This work was supported by JST Presto and JSPS KAKENHI Grant Number 16H01721.

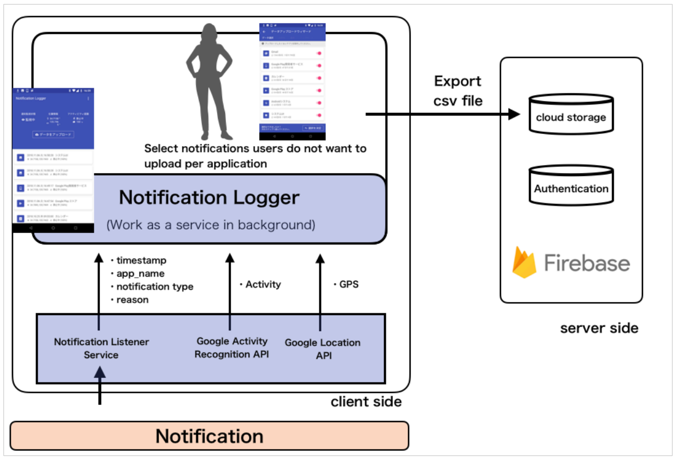

Fig. 1. system architecture

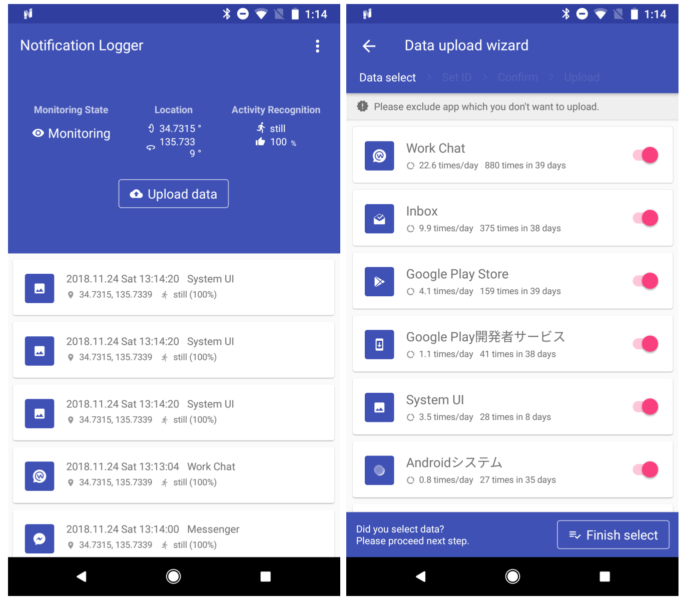

Fig. 2. Screenshots of Notification Logger

on use-side rather than providing the same service to all mobile users as in the past. Ho et al. have proposed a personalization method of notification timing based on reinforcement learning using user context as a method for optimizing notification timing for each individual [6].

C. NotificationListenerService API

On Android terminals we can use API called `NotificationListenerService` [7]. developers can receive information such as notification application name, text message, timestamp. when the user receives or deletes the notification on the smartphone by using this API. Weber et al. have developed an open source framework that more than 60,000 users will use for notification research on mobile devices [8]. In addition, Sahami et al. evaluated notifications of message applications including information on users and events, and performed notification analysis focusing on the subjectivity of users [9].

D. Purpose of this study

In this research, in order to realize adaptive notification management on the use-side, i.e., maximize the response rate for each notification sent to the user, we consider a method of controlling the timing adapted to the real-time context on the use-side and the response situation to all applications. The position of this paper is to investigate the diversity of the notification received by the user and the behaviors to all notifications in the smartphone, and clarify the feasibility of adaptive notification management.

III. Survey System

A. System architecture

Figure 1 shows the system architecture of our system, named *Notification Logger*. And, Figure 2 shows screenshots of Notification Logger. In this experiment, we first acquire the behavior of receiving or deleting notifications of all the applications that the user is permitted to notify through `NotificationListenerService` running in the background. Then, the behavior and location information of the users at that time is acquired from the `ActivityRecognitionAPI` [1] and the `FusedLocationProviderAPI` [2], and stored in the smartphone. After that, it adopts a mechanism to export the log data saved manually by the user. The major difference from the related research mentioned in Section II-C is that it supports logging of notification actions (e.g., notification click action, notification delete action, notification delete action for the same application) which can be acquired from the latest OS of Android 8.0 or later. Therefore, it is possible to log actions of notification that the user opened the notice and confirmed the contents, while it was only able to confirm that the notification was erased in conventional systems. In this survey, we analyze mainly the opening operation of notifications by users.

B. Attention to privacy

Table I shows data collected by the system. By using the `NotificationListenerService`, we acquire the timestamp when receiving or deleting the notification, the application name included in the notification, *notification type* (receive or delete the notification), and *notification action* (the operation performed by the user). In addition, user location information at that time was acquired from the `FusedLocationProviderAPI` and user actions were acquired from the `ActivityRecognitionAPI`.

Although text messages included in the notification can be originally acquired, in this experiment, we did not intentionally acquire the contents of the message in order to lower participating hurdles of the users. Nevertheless, they still include personal information data such as the application name, position information, action information. That is why

[1] https://developers.google.com/android/reference/com/google/android/gms/location/ActivityRecognitionApi

[2] https://developers.google.com/android/reference/com/google/android/gms/location/FusedLocationProviderApi

TABLE I
COLLECTED DATA

Data	Value
Timestamp	Time when notification action occurred
Location information[*a]	GPS Information (Latitude, Longitude) when notification action occurred
App name[*b]	Application name of notification action
Notification type[*b]	Posted Removed
Notification action[*b] (Reason code)	REASON_CLICK (1)[*c] REASON_CANCEL (2, 8)[*d] REASON_APP_CANCEL_ALL (9)[*e] REASON_GROUP _SUMMARY_CANCELED (12)[*f]
Activity recognition[*g]	IN_VEHICLE ON_BICYCLE ON_FOOT STILL TILTING UNKNOWN

[*a] Use `FusedLocationProviderAPI`
[*b] Use `NotificationListenerService` [*c] Notification click action
[*d] Single notification delete action. In this research, REASON_CANCEL (2) and REASON_APP_CANCEL (8) are considered to be the same
[*e] Group notification delete action [*f] All notification delete action
[*g] Use `ActivityRecognitionAPI`

TABLE II
EXPERIMENT PARTICIPANT ATTRIBUTE

			Number
Breakdown of participant attribute	Gender	Male	7
		Female	13
	Age	20–29	8
		30–39	9
		40–49	3
	Profession	Employee	9
		Housewife	6
		Part-time job	3
		Student	2
Total number			20

TABLE III
RESULT OVERVIEW

	Value	Result
Overview	Total number	95,910
	Received count	68,261
	delete count	27,649
Breakdown of notification action	Notification clickopening	3,506
	Single Notification delete	17,634
	Group Notification delete	3,643
	All Notification delete	2,769
Breakdown of app type	Notification from message apps	22,088
	Notification from money apps	1,908
	Notification from game apps	337
	Notification from news apps	434
	Notification from map apps	3,456
	Notification from OS apps	5,835
	Notification from others	34,201

we developed the system which does not automatically upload the log data to the server but store them in the smartphone for protecting user's privacy. The users themselves can check the log data before uploading data in this system.

IV. Survey experiment

A. Investigation method

In this experiment, we recruited experiment participants using "Crowdworks" which is a major Japanese crowdsourcing service. We conducted data collection experiments for 20 people in the applicants for four weeks. Participants installed the "Notification Logger" mentioned in Section III on the Android smartphone normally used and gave permission to acquire the position information and the sensor information, and in the state where notifications are being monitored and the spent usual life.We sent reminds to participants every week and uploaded data in 4 times.

B. Investigation conditions

We adopted only applicants who are using terminals equipped with Android 8.0 and above because `NotificationListenerService` used in Notification Logger can get the behavior of notification tap (Click, Opening) in only Android OS version of Android 8.0 or higher. Table II shows the participant's attributes. Participants targeted 20 males and females up to the age of 20–49.

V. Results and discussion

Table III gives an outline of the experiment result. In this experiment, 95,910 notification data could be acquired in four weeks. In the breakdown of the notification data, the number of notifications received was 68,261, and the number of notifications deleted was 20,345. The 3,506 cases of delete are deleted by notice click (opening), the 17,634 cases are deleted by user-initiated single notification, the 3,643 cases are deleted by group deleting function by application, and the 2,769 cases are deleted by all delete function of notification.

Also, we grouped apps with their main function as follows: *message app*, which are communication apps such as chat, mail and SNS; *money app* such as crowdsourcing and flea market apps; *game app* such as social game apps; *news app* such as Yahoo news and Gunosy; *map app* such as Google Maps; *OS apps*, which is the system apps of Android OS; and *others*.

Many data included notifications from the OS apps, however, there were many phenomena that on the smartphone it was not displayed as a notification, or the `NotificationListenerService` caught the log consecutively as if it received multiple notifications during the downloading operation of the apps etc. Therefore, in this paper, data from the OS apps are removed and analyzed.

A. Analysis result in perspective of app type

We consider the *click through rate (CTR)*, which is the ratio of click actions to all posted notifications, for each application type and the response time until the notification click action.

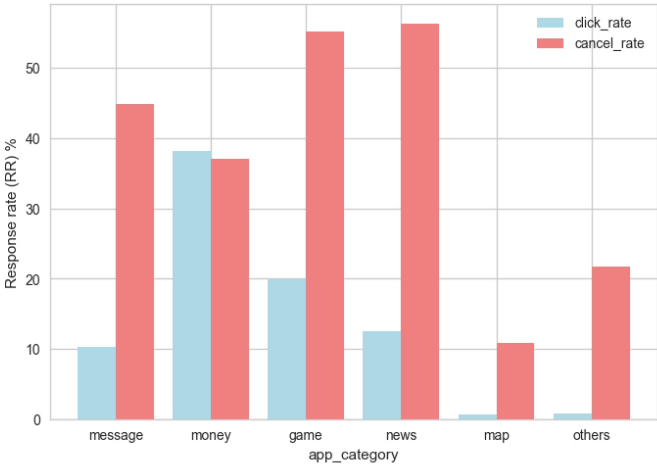

Fig. 3. Response rate (RR) for each application type.

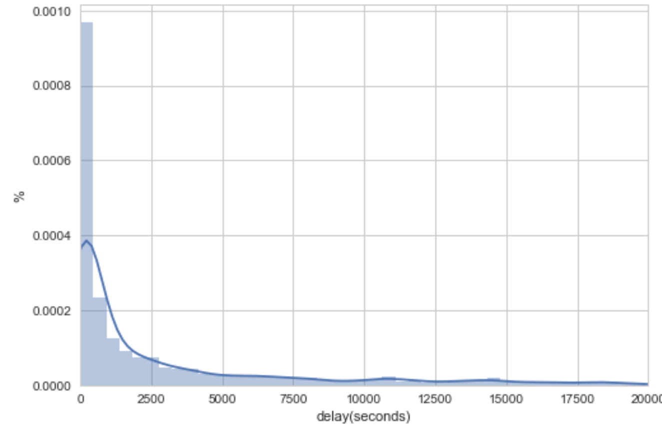

Fig. 4. Histogram of response time until the notification click action.

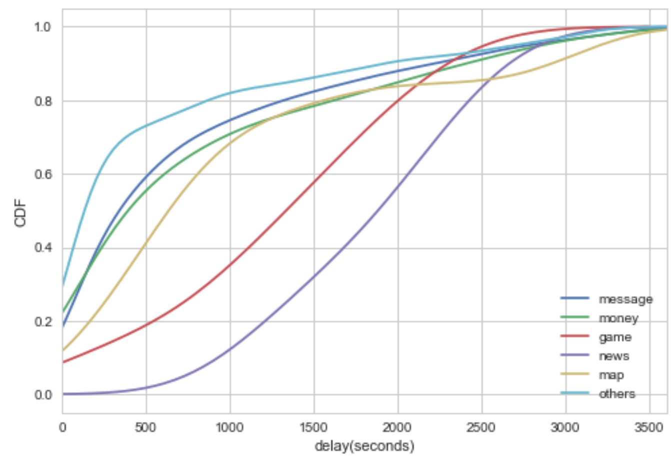

Fig. 5. CDF on users' response time to notifications.

1) Response rate for each app type: Figure 3 shows *response rate (RR)*, which is the ratio of click and delete actions to all notifications, for each app type. The CTR of the money app is the highest, the next is the game app, the news app, the next, except the others, the CTR of message app with the largest number of notifications was the fourth. Similarly, when comparing with the notification delete rate, the money application has a lower delete rate than the message, the game, and the news application, and it is found that it is a category of the application that the notification is easily accepted for users.

2) Response time until click action for each app type: We consider based on the response time from receiving the notification until the application is clicked.

Figure 4 shows a histogram of the response time from the reception of the notification until it is clicked. From the figure, you can see that most of the notifications that clicks are being clicked within about 1 hour (3600 s) since it was received. Hence, we analyze the notification clicks which are responded within 1 hour.

Figure 5 shows the *cumulative distribution function (CDF)* of response time until notification click for each type of application. The notifications from others resulted in the shortest response time, however as can be seen from Figure 3, the RR was extremely low. It is conceivable that the graph is shifted to the left due to the fact that the click data is small overall. Also, from the graph, it was found that the money application whose the RR was the highest in Figure 3 has a longer response time than the message application, and except for others, the message application is the category of the application which is the easiest to click on.

3) Inducing of click action by succeeding notifications: As shown in Figure 6, the click event by the user is not necessarily caused by the notice of the application itself. For example, we should also consider that the user can click the notification, which the user already received from "app y" before, at the timing of receiving the notification from "app x."

Table IV shows the aggregated results the number of notifications when the last received notification is sent from the same application and different applications for one notification clicked by the user. In the message application, as can be seen from the response time, a majority of notifications instantaneously induce their own click behavior. However, most of money, news, and game apps received notifications from other applications before, it turned out that there were quite a lot of cases where notification reception timing of the application did not directly affect clicking behavior.

B. Analysis result in perspective of spot

1) Click through rate for each spot: In order to analyze user's position information for each fixed range, conversion from the acquired latitude and longitude to a scale called *GeoHex* was performed. This is a scale to fill all over the world with Hex (regular hexagon) without gaps and to express all the points in the world [10]. In this study, we converted the level of GeoHex to Level 6 which makes the center distance about

TABLE IV
RELATIONSHIP BETWEEN CLICKED NOTIFICATION AND ITS JUST BEFORE NOTIFICATION.

App type	Clicked notification and its just before notification is:	
	same app	**different app**
Message	1,172	941
Money	156	503
News	0	49
Game	4	52
Map	6	15
Others	31	25
Total	**1,369**	**1,585**

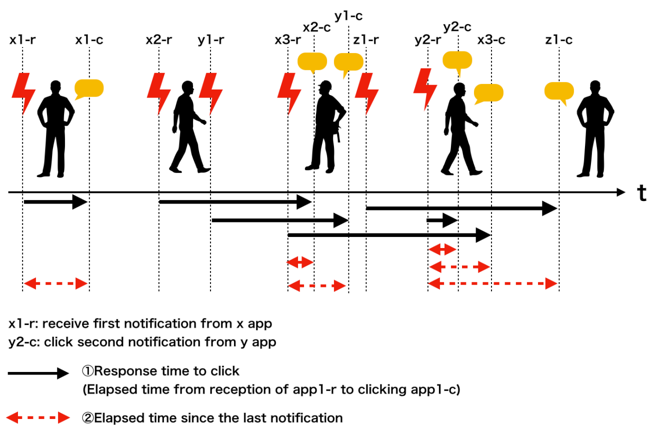

Fig. 6. Response time since the last notification.

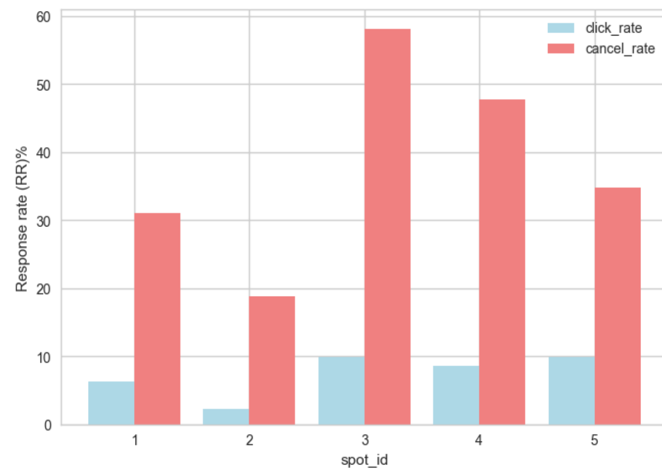

Fig. 7. Response rate at the top five spots.

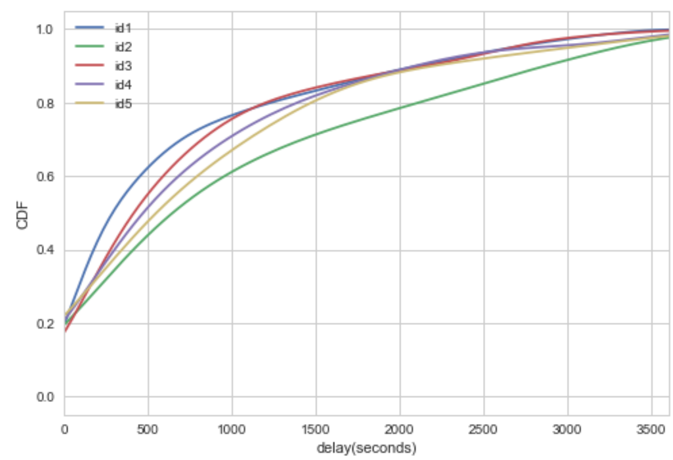

Fig. 8. CDF on users' response time to notification at the top five spots.

2.7 km using the GeoHex python library [3]. We sorted the ID as the spot where the user stayed frequently in descending order of the number of data with respect to the spot after conversion. The id1 represents the spot where each user stayed most frequently, id2 represents the spot the second most frequently stayed. Figure 7 shows the RR at the top five spots of the stay frequency. From the graph, it can be seen that the CTR is the lowest despite the place where id2 is staying the second frequently among all the spots.

2) Response time per spot: The CDF of response time until the notification click at each spot is shown in Figure 8. It was found that the spots that stay the second most frequently for the user are spots that are most difficult to respond to notifications, since id2 has the longest response time to click at the spot.

3) Relationship between users and spots: Figure 9 shows the relationship between the spot with the top 5 stay frequencies and the time slot. From the distribution of spots by time slot, it is seen that id1 is a spot often staying at midnight and id2 is statistically a spot where the user frequently stays during the day time. Therefore, it can be considered that id1 is likely to be the user's home, id2 is likely to be the workplace or school for the user.

[3]https://pypi.org/project/py-geohex3/

C. Analysis result in perspective of time slot

Figure 10 shows the CTR per time slot. It can be seen that The number of notifications received has increased from 12 o'clock in the day and from 18 o'clock to 20 o'clock, however, there is not much change in the number of clicks except for bedtime. Therefore, there might be an upper limit as to the number of notifications the user can click at each time, and it can be considered that there is a high possibility that the CTR will be lowered even if notification is sent over the number of upper limit.

VI. FOR ADAPTIVE NOTIFICATION MANAGEMENT

In this study, responses of notifications were statistically analyzed for category classification of applications, reception spot classification according to stay frequency, and reception time slot classification, and the following results were confirmed.

- Categorization of applications is an effective index to predict responsiveness of notifications in adaptive management.

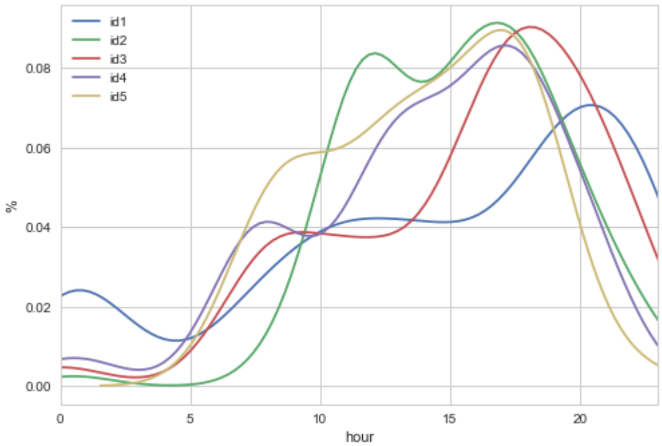

Fig. 9. Distribution of the spot for each time slot.

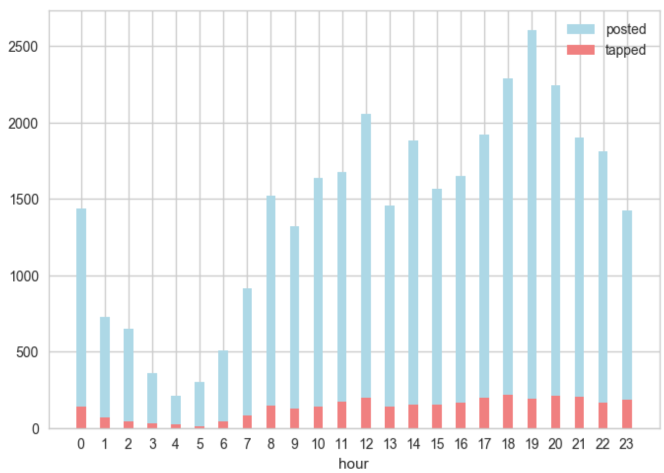

Fig. 10. Number of receiving and clicking of notifications for each time slot.

- Statistically, the spot staying the second most frequently for the user has a high correlation with responsiveness.
- It was confirmed that there is an upper limit to the number of responses to notifications per unit time.

In addition, from Table IV, there are still few cases where the notification timing of the application is directly connected to the click, so it can be considered that there are many cases induced by other notifications, so it is possible to consider not only a single application, by clarifying the correlation between the categories of applications that have been opened and opened applications in the future, there is a possibility to propose a control method to more effectively induce notification responses.

VII. Conclusion

In this research, in order to realize adaptive notification timing control, we developed a system that can safely investigate notifications received by users, and investigated the diversity of all notifications and the behavior of users for each. We have collected 95,910 notifications data from 20 participants recruited by crowdsourcing through four weeks of survey experiments, and statistically analyzed them.

The rate of notification responses to applications related to money was the highest, We found that the message type application is the fastest response category. In addition, it was confirmed that the notification responsiveness was greatly decreased at the spot that was staying the second most frequently for the user, and it was confirmed that the number of responses to human notifications still had an upper. In this research, it became newly discovered that many notification opening actions were induced by notification of other applications. By clarifying the correlation between notifications, we aim to realize an adaptive notification timing control system on the terminal side that is close to the user.

References

[1] T. Okoshi, K. Tsubouchi, M. Taji, T. Ichikawa, and H. Tokuda, “Attention and engagement-awareness in the wild: A large-scale study with adaptive notifications,” *2017 IEEE International Conference on Pervasive Computing and Communications, PerCom 2017*, pp. 100–110, 2017.

[2] D. Weber, A. Voit, P. Kratzer, and N. Henze, “In-situ investigation of notifications in multi-device environments,” *Proceedings of the 2016 ACM International Joint Conference on Pervasive and Ubiquitous Computing - UbiComp ’16*, pp. 1259–1264, 2016. [Online]. Available: http://dl.acm.org/citation.cfm?doid=2971648.2971732

[3] A. Mehrotra, M. Musolesi, R. Hendley, and V. Pejovic, “Designing content-driven intelligent notification mechanisms for mobile applications,” *Proceedings of the 2015 ACM International Joint Conference on Pervasive and Ubiquitous Computing - UbiComp ’15*, pp. 813–824, 2015. [Online]. Available: http://dl.acm.org/citation.cfm?doid=2750858.2807544

[4] E. Cutrell, M. Czerwinski, and E. Horvitz, “Notification, disruption, and memory: Effects of messaging interruptions on memory and performance,” *Proceedings of the INTERACT 2001*, no. 1999, pp. 263–269, 2001.

[5] G. Mark, S. T. Iqbal, M. Czerwinski, P. Johns, A. Sano, and Y. Lutchyn, “Email Duration, Batching and Self-interruption,” *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems - CHI ’16*, pp. 1717–1728, 2016. [Online]. Available: http://dl.acm.org/citation.cfm?doid=2858036.2858262

[6] B.-j. Ho, L. Angeles, M. Koseoglu, and L. Angeles, “Nurture : Notifying Users at the Right Time Using Reinforcement Learning,” 2018.

[7] “Android developers. 2018. notificationlistenerservice. https://developer.android.com/reference/android/service/ notification/notificationlistenerservice.html.”

[8] D. Weber and A. Voit, “Notification Log : An Open-Source Framework for Notification Research on Mobile Devices,” pp. 1271–1278.

[9] A. Sahami Shirazi, N. Henze, T. Dingler, M. Pielot, D. Weber, and A. Schmidt, “Large-scale assessment of mobile notifications,” *Proceedings of the 32nd annual ACM conference on Human factors in computing systems - CHI ’14*, pp. 3055–3064, 2014. [Online]. Available: http://dl.acm.org/citation.cfm?doid=2556288.2557189

[10] @sa2da, “GeoHex - Hexagonal Geo-coding System,” https://sites.google.com/site/geohexdocs/.