ABSTRACT

Prior HCI studies often analyzed smartphone app usage data for usability and user experience research purposes. App usage videos are often collected by a screen recording app in order to better analyze users’ app usage behaviors (e.g., app usage time, screen transition, and notification handling). However, it is difficult to analyze app usage videos along with multiple user interaction stream data. When the length of a video is long, data analysis tends to take a long time due to the volume of user interaction data. This is even more difficult for novice researchers due to a lack of data analysis experience. In this paper, we propose LV-Linker (Log and Video Linker), a visualization tool that helps researchers quickly explore the app usage log and video data by linking multiple time series log data with the video data. We conducted a preliminary user study with eight participants to evaluate the benefits of linking, by measuring task completion time, helpfulness, and subjective task workload. Our results showed that offering a linking feature significantly lowers the task completion time and task workload.

Keywords Mobile, App Usage Log, Screen Recording, Video Analysis, Time-series Data, Log Analysis, App Usage Behavior, App Usage Pattern, Smartphone Use

1 Introduction

Smartphone app usage data collected through user-smartphone interaction is utilized in various industries and HCI research fields. Representing, UI/UX designers use app usage data-driven UI/UX design for smartphone home screen and app UI/UX enhancement. Furthermore, the app usage data is being used in many user-centric computing research fields (e.g., digital phenotype, understanding user’s smartphone usage pattern, improving user’s interaction & accessibility, enhancement of UI/UX) [1, 2, 4, 5, 6, 7, 8].

The recorded app usage video data can be used to understand the user’s app usage behavior patterns. There are two ways to collect app usage video data: (1) using a smartphone system built-in screen recording app or third-party screen recording app, and (2) collecting through a wearable camera. However, in the case of Android and iOS, which are representative smartphone OS, Android 10 released in September 2019 and iOS 11 released in June 2017 supported their screen recording function, respectively [9, 10]. Therefore, previous studies published before 2019 identified app usage behavior patterns by collecting app usage video data through an independently developed screen recording app. 

LV-LINKER: SUPPORTING LINKED EXPLORATION OF PHONE USAGE LOG DATA AND SCREEN VIDEO DATA

A PREPRINT

Hansoo Lee
School of Computing
KAIST
Daejeon, Republic of Korea
hansoo@kaist.ac.kr

Sangwook Lee
School of Computing
KAIST
Daejeon, Republic of Korea
sangwooklee@kaist.ac.kr

Youngji Koh
School of Computing
KAIST
Daejeon, Republic of Korea
youngji@kaist.ac.kr

Uichin Lee
School of Computing
KAIST
Daejeon, Republic of Korea
ulee@kaist.ac.kr

May 31, 2022
This app usage video data shows detailed app usage behavior patterns compared to app usage data. This app usage log data contains text, numerical, and categorical data to identify users’ app usage behavior patterns. These app usage log data can be collected by the app usage logger developed by Android/iOS built-in APIs provided for third-party app developers such as UsageStatsManager and AccessibilityService in Android OS. Aware Framework is one of the representative app usage loggers for app usage analysis. These app usage loggers can collect the app usage log data such as app usage status (foreground/background/stop time, app & package name, screen transition), view hierarchy/elements, key typing, notification, and touch interaction type (e.g., click, long click, swipe, scroll).

App usage log data has many advantages in terms of research to identify app usage behavior patterns compared to app usage video data. This app usage log data includes touch interaction information that occurs when a user interacts with a smartphone, which cannot be identified with app usage video data. In addition, app usage log data, which is time-series data of numerical, categorical, and text types of data, is easy to remove sensitive data through preprocessing. Therefore, using app usage log data for research is easier to protect privacy issues than using app usage video data. Furthermore, even if the app usage log data is collected for a long time, the capacity is relatively small compared to the app usage video data, and it is possible to quickly find and analyze a specific point in time through timestamp information on the spreadsheet. In addition, app usage log data has the advantage of being easier to perform data analysis (e.g., statistical analysis and machine learning) compared to app usage video data. Therefore, since app usage log data can accurately identify app usage behavior patterns in real-time, it is currently used in various research fields more than app usage video data to quantitatively analyze app usage behavior patterns.

However, to collect app usage log data, an app usage logger must be developed using Android built-in APIs, and there are various data types and attributes as well. Therefore, there is a learnability issue in collecting app usage log data and using it for research. In this study, we propose an LV-Linker which is Log and Video Linker, a viewpoint movement interaction system that synchronizes the time of app usage video and logs data, and moves the viewpoint of one data point as well as the viewpoint of another data. LV-Linker is a system configured by combining a spreadsheet-based log viewer for app log data and a video player-based video viewer for app video data. Through this research, we expect that the proposed system help researchers to analyze the app behavior patterns more easily, quickly, and accurately than when they used both types of data separately.

2 Related Work

Existing studies that connect time-series log and video focus on linking video and animation or text for audio description of the video. These studies have in common with our study in that they can move the viewpoint through the interaction between video and text to analyze or find important viewpoints with annotations or text-connected timestamps. However, those studies are different from our study in that they did not use time series-based app usage log data, and they are not targeting a mobile environment.

The previous study in a mobile environment is to create a system replayer that can summarize statistical data or focus on specific factors of interest by synchronizing images taken while using mobile systems and logs generated from the system. And there is a systematic study that allows users to check wearable cameras in which surrounding environments are recorded for user experience recording when using smartphones. However, in existing studies, the video was not filmed on a mobile screen, but filmed from afar with a camera, and rather than focusing on the connection with app usage data, it was designed to view multiple people’s mobile system logs and usage videos at the same time and check the user’s context. Therefore, this study differs from previous studies in that it focuses on connecting recorded videos using Android app usage logs and recorded apps inside smartphones.

Therefore, this study proposes LV-Linker, a system that connects screen recording video data and time series app log data that can support app usage analysis in terms of data understanding, correctness, and quickness in a mobile environment. Accordingly, the purpose of this study to understand the linking effect of app usage log and video data is
LV-Linker: Supporting Linked Exploration of Phone Usage Log Data and Screen Video Data

As follows: (1) We compare the performance of the proposed system and app logs/recorded videos in terms of exploring app usage data. (2) We examine how the prior experience of analyzing app usage affects the use of the proposed system compared to app logs and recorded video. (3) We examine how LV-Linker helps to perform app usage data analysis tasks compared to app logs and recorded videos in terms of correctness, quickness, and data understanding.

3 LV-Linker: Linking Usage Log & Video

3.1 System Description

We developed LV-Linker to help researchers evaluate how linking app usage logs and videos affects users’ correctness, quickness, data understanding, and experience when analyzing smartphone app usage data. Log viewer shows the processed logs in the form of a spreadsheet as shown in Figure 1(a). The video player shows the app usage video that records users’ app usage behavior pattern on the smartphone in Figure 1(b). It supports a basic seek bar to change into a specific frame. The users can select the video with the dropdown menu as depicted in Figure 1(c). LV-Linker integrates the log viewer that shows the processed logs in a spreadsheet into the video player that shows user behavior on the smartphone as depicted in Figure 1. LV-Linker added a toggle switch to measure the effects of linking only to link or unlink the log and video in the same UI as shown in Figure 1(d). Furthermore, the system embedded the task sheet for participants to immediately copy and paste the required analysis results as depicted in Figure 1(e). The system was implemented and deployed on the web frontend based on TypeScript and React.

3.2 LV-Linker Usage Scenarios and Interaction Examples for Evaluation

To showcase the benefits of LV-Linker, we consider the following data analysis tasks that could be used in the usability evaluation experiments. Twelve videos were recorded by a smartphone screen recording app at different times. Each video contains 6-minute app usage behaviors of the following five app tasks which are the most frequently used smartphone apps [22, 23]. For the following smartphone app tasks, we created and recorded data on our own, such as making calls and sending text messages using two smartphones using virtual names.

- **Answering a call**: Leave a different message depending on the person who called.
- **Sending SMS**: Send a reply to a message that makes an appointment for a time, place, and date.
Table 1: Description of filtered app usage log data used for usability evaluation experiments of LV-Linker System

| Datum Type          | Column   | Data Type | Description                                      |
|---------------------|----------|-----------|--------------------------------------------------|
| APP_USAGE_EVENT     | Type     | Activity_paused | App moves to the background                     |
|                     | Activity_resumed |           | App moves to the foreground                      |
|                     | Activity_stopped   |           | App activity is terminated                       |
|                     | Name      | string     | User expression name of the app in the current event state |
|                     | packageName | string     | Developer package name of the app in the current event state |
|                     | timestamp | numerical  | Timestamp for the current event state            |
| KEY_LOG             | currentKey | string     | The currently entered key raw value             |
|                     | timeTaken  | numerical  | Time difference between prevKey and currentKey input |
|                     | Name       | string     | User expression name of the app in the current key value |
|                     | packageName | string     | Developer package name of the app in the current key value |
|                     | timestamp  | numerical  | Timestamp for the current key value             |
| NOTIFICATION        | isPosted   | boolean    | Whether the notification is currently posted or not |
|                     | Name       | string     | User expression name of the app in the current notification event |
|                     | packageName | string     | Developer package name of the app in the current notification event |
|                     | timestamp  | numerical  | Timestamp for the current notification event     |

- Taking pictures: Take pictures of business cards in the camera app and delete one of them in the gallery app.
- Sharing a route: Find a route to a specific store in Google Maps and share the route link through a message.
- Transferring money: Transferring money to the account provided in a message in a banking app.

We ordered these videos randomly and placed them within the dropdown menu so that participants can move on to the following video for every task as depicted in Figure 1(c).

We used a self-developed app usage logger by Android built-in APIs [14, 15, 24, 25] to collect the app usage logs such as the app usage status (e.g., start & end time, app & package name), notification (e.g., posted, app & package name), and typing event (typed letters & time, app & package name), touch interaction type (e.g., click), device event (screen on/off, battery status, power on/off), call & SMS logs. The app usage logs consist of metadata such as timestamp, datum type, and JSON data with various attributes (e.g., package name, app name, type, is posted). Our system appends metadata to the parsed JSON data and presents the data on the sheet.

As shown in Figure 1(f), Users can select the data they want to see using the filter function on the Log viewer. The filter consists of two stages: (1) Show/Hide columns, (2) Filter by the value of the selected column. The columns are grouped by datumType, which is the type of app usage log data (e.g., app usage event, notification, key log, device event) because each datumType has different sets of columns. To simplify the task for the experiment, we used a filter feature to exclude useless columns and datum types (e.g., device event, call & SMS, touch interaction type). The remaining logs belong to APP_USAGE_EVENT, KEY_LOG, and NOTIFICATION datumType. The remaining columns are timestamp, datumType, type (e.g., app usage interaction type), name, package name, currentKey, timeTaken, isPosted.

Table 1 has detailed descriptions and data types about datumType and columns of app usage log data used in LV-Linker.

The app usage log and video data are synchronized based on the timestamp. Thus, when users select a specific frame on the video player, the log viewer highlights a log recorded at the same frame. Conversely, when the users select a specific log on the log viewer, the video player seeks the corresponding time frame on the video player. Then, we manually synchronized the log and video based on one key typing event that we can check in both. Thus, there is a slight sync error between them.

### 4 Preliminary Evaluation

#### 4.1 Evaluation Setup and Procedure

We recruited eight university students (5 female, 3 male) through online community posting. Through the preliminary survey, we divided participants into two groups: experienced and non-experienced people. Only two of them have prior experience in analyzing app usage data. The experiment was 90 minutes long. The research was exempt from the Institutional Review Board (IRB) under the condition that we do not collect participants’ personal information.

The experiment focused on exploring the effects of linking logs and videos when analyzing app usage data. Before experimenting, participants were asked to perform a preliminary survey. The preliminary survey asked questions about prior knowledge of Android app usage log data, and smartphone information (model name, OS) they are using. Through this survey, participants were classified into experienced groups and non-experienced groups according to prior knowledge of app usage data.
LV-Linker: Supporting Linked Exploration of Phone Usage Log Data and Screen Video Data

A PREPRINT

Figure 2: Task Description (a) App Keyboard typing analysis (b) The number of App transition analysis (c) App usage start/end time analysis in different time (d) App usage start/end time analysis in continuous time

Then, we explained the LV-Linker system and the tasks the participants should perform. Four tasks that participants should perform to analyze app usage data. Tasks are selected based on the app usage data analyzed in previous studies [26, 27, 28, 29]. In the first task, they should find typing information entered by text in the SMS app and find a time interval between two entered specific letters. In the second task, they should find the number of app transitions between Camera & Gallery app. In the third task, there are two videos recorded at different times and log data which includes logs in two videos. In the first video, they find the time when the notification was posted and when the message was shared on Google Map. In the second video, they find the time when the notification was unposted and when the SMS app ended. In the last task, they find when an app started/resumed and ended/paused in the banking app. The detailed description of each task is shown in Figure 2.

In the main experiment, participants were divided into two groups (group A and B) according to the order of use of the system (linked, unlinked) and performed the above four tasks. Group A used the unlinked system after using the linked system, and group B are in reverse order. The total experiment was 90 minutes long. In the first session, which is a practice session, participants learned how to use the system and basic knowledge of app usage data. We asked for the same four tasks for each session so that participants work on two iterations (practice, main). For quantitative evaluation, we measured the task completion time and the accuracy of task performance. Task completion time was measured every time participants finished each task, and when they finished one session, researchers recorded their time.

Last, the participants were asked to perform a post-survey. The post-survey asked questions about the helpfulness of the linked system for four tasks as 5 Likert scales from three perspectives: 1) correctness, 2) quickness, and 3) data understanding. We measured cognitive load using NASA Task Load Index (NASA-TLX) when analyzing app usage data with linked and unlinked systems, respectively: Mental/Physical/Temporal Demand, Effort, Performance, Frustration. Last, we asked for user feedback about the system.

4.2 Result and Findings

To determine statistical analysis methods, we examined the normality of Kolmogorov-Smirnov and Shapiro-Walk for quantitative and qualitative evaluation results. To analyze the result between the within-subject and the between-subject that is repeated differences for the results in the non-parametric statistics, the two-way mixed ANOVA repeated messages...
Table 2: Two-way mixed ANOVA RM results on task completion time for within linked/unlinked system and between expert/novice group or A/B group

| Task Time  | Source  | d.f | F value  | P     | Source  | d.f | F value | P     |
|------------|---------|-----|----------|-------|---------|-----|---------|-------|
| Overall Task | L/U     | 3   | 5.156    | 0.01103* | L/UL    | 3   | 5.628   | 0.00789** |
|             | E/N     | 1   | 9.948    | 0.00614** | A/B     | 1   | 2.612   | 0.12558 |
|             | L/U * E/N | 3   | 0.467    | 0.709   | A/B * L/UL | 3   | 3.746   | 0.03266* |
|             | error   | 16  |          |        | error   | 16  |          |        |
| Task 1      | L/U     | 3   | 1.525    | 0.246  | L/UL    | 3   | 1.822   | 0.152  |
|             | E/N     | 1   | 2.028    | 0.174  | A/B     | 1   | 2.946   | 0.105  |
|             | L/U * E/N | 3   | 0.331    | 0.803  | A/B * L/UL | 3   | 3.746   | 0.143  |
|             | error   | 16  |          |        | error   | 16  |          |        |
| Task 2      | L/U     | 3   | 6.388    | 0.00473** | L/UL    | 3   | 7.488   | 0.00237** |
|             | E/N     | 1   | 4.712    | 0.04534* | A/B     | 1   | 0.329   | 0.57445 |
|             | L/U * E/N | 3   | 0.521    | 0.67369 | A/B * L/UL | 3   | 3.262   | 0.04903* |
|             | error   | 16  |          |        | error   | 16  |          |        |
| Task 3      | L/U     | 3   | 4.409    | 0.0193* | L/UL    | 3   | 3.612   | 0.0348* |
|             | E/N     | 1   | 8.319    | 0.0108* | A/B     | 1   | 1.003   | 0.3314 |
|             | L/U * E/N | 3   | 0.276    | 0.8419 | A/B * L/UL | 3   | 0.343   | 0.7944 |
|             | error   | 16  |          |        | error   | 16  |          |        |
| Task 4      | L/U     | 3   | 1.987    | 0.157  | A/B Session | 1   | 1.009   | 0.3301 |
|             | E/N     | 1   | 1.043    | 0.322  | L/UL    | 3   | 2.711   | 0.0797 |
|             | L/U * E/N | 3   | 0.259    | 0.854  | A/B * L/UL | 3   | 2.432   | 0.1028 |
|             | error   | 16  |          |        | error   | 16  |          |        |

(RM) and turkey post hoc are used. In quantitative evaluation results, the task completion time of an experienced and a non-experienced person was not properly collected, so these two participants’ data were excluded.

4.2.1 Task Completion Time

First, a two-way mixed ANOVA RM test was performed between expert/novice groups and on each of the five tasks (Task 1, 2, 3, 4, Overall task) completion times for the within four-session (practice and main session with linked/unlinked system). As shown in Table 2, we found that there was a significant difference within four-session and between expert/novice groups in the total task, tasks 2, and 3 (p<0.05). However, we found that there were no significant interaction effects of the two factors.

Second, a two way mixed ANOVA RM test was performed for five task completion times between group A/B and within the four-session, respectively. As a result, we found that there was no significant difference between groups A/B in all five task completion times. As before, we analyzed that there was a significant difference between within-subjects in the total task, task 2, and 3 (p<0.05) in Table 2. However, we found that there was a significant interaction effect of the two factors which are the group A/B and session in the total task and task 2 (p<0.05) in Table 2.

In group A, the main session of task completion time was much shorter compared to the practice session. Accordingly, the linked system can help users to understand the log data in hard levels, so the learning effect was high when the users used the linked system first. The difference in task completion time within linked/unlinked systems and between expert/novice groups was significant in Task 2, 3, not Task 1, 4. Because Task 1, 4 is easier (see more video than log) than Task 2, 3. Therefore, the more difficult the task is, the more effective the linked system is than the unlinked system.

4.3 Helpfulness & Task Workload

In the helpfulness survey, 75% of respondents answered that the linked system was more helpful than the average in the overall task in terms of data understanding, correctness, and quickness compared to the unlinked system in Figure 3. In each data understanding, correctness, and quickness, they answered that the linked system was more helpful...
than the unlinked system in Tasks 3 and 4 in Figure[3]. As a result of two-way within ANOVA (five tasks: Task 1, 2, 3, 4, Overall task; survey items: data understanding, correctness, quickness), a significant difference only occurred for five task factors (p<0.05). There was no interaction effect between the two factors. Furthermore, two-way mixed ANOVA analysis between the expert/novice group and within the five tasks, there was no significant difference, and no interaction effect as well.

In the NASA TLX survey, the average load reduction of the linked system compared to the unlinked system of NASA TLX among all participants in the experiment was performance (42%), physical demand (35%), temporal demand (31%), effort (0.30%), mental demand (24%), and frustration (20%) showed a reduction of more than 30% in the four categories as shown in Figure[4]. Users responded that the linked system reduced the user’s load by 30% or more compared to the unlinked system in the performance, physical/temporal demand, and effort items. Moreover, we found that there was a significant difference in all items except for the frustration item between linked/unlinked systems in NASA TLX (p<0.05). We found that there was a significant difference between expert/novice groups in mental demand items by two-way mixed ANOVA RM for each survey item between expert/novice groups and within a linked/unlinked system in NASA TLX (p<0.05), but no interaction effect between the two factors. We found that the linked system had a lower load compared to the unlinked system in both expert and novice groups. In helpfulness, the linked system is to reduce the load of users in quickness, data understanding, and correctness in performing app usage analysis tasks.

4.4 Subjective User Experience Reports

We received short answers on the advantages & disadvantages of linking log & video and system improvements from all participants. Participants responded that linking was very helpful for quickness and data understanding. Non-experienced participants said that it was good to intuitively know which data is related to which activity. Experienced participants said that the location where the log starts can be immediately found through the video, so the action of scrolling to the corresponding part is omitted, which is likely to increase speed. However, some participants expressed concern about the system. They responded that if there is abnormal data, it is not sure which data (log or video) to believe. It can be more confusing than when using an unlinked system. In addition, they answered that it was difficult to
accurately pause the specific point in the video because the event passed in an instant. As a system improvement, they responded that it would be good to add a function to automatically pause the video when certain log data is selected during video playback, and a function to label log event information on the video progress bar.

5 CONCLUSION AND FUTURE WORK

This study verified the effectiveness of app usage analysis through quantitative and qualitative evaluation results through LV-Linker, a system that proposed the linking effect of app usage log and video. The proposed LV-Linker system is expected to help analyze the app usage behavior pattern in various mobile and HCI studies using app usage logs and video data. In future work, our proposed system can link log data on the web or other smart devices (e.g., smartwatch) and video (display information) to analyze user interaction data.

References

[1] Mehdi Boukhechba, Yu Huang, Philip Chow, Karl Fua, Bethany A Teachman, and Laura E Barnes. Monitoring social anxiety from mobility and communication patterns. In Proceedings of the 2017 ACM International Joint Conference on Pervasive and Ubiquitous Computing and Proceedings of the 2017 ACM International Symposium on Wearable Computers, pages 749–753, 2017.

[2] Karen Church, Denzil Ferreira, Nikola Banovic, and Kent Lyons. Understanding the challenges of mobile phone usage data. In Proceedings of the 17th International Conference on Human-Computer Interaction with Mobile Devices and Services, pages 504–514, 2015.

[3] Uichin Lee, Joonwon Lee, Minsam Ko, Changhun Lee, Yuhwan Kim, Subin Yang, Koji Yatani, Gahgene Gweon, Kyong-Mee Chung, and Junehwa Song. Hooked on smartphones: an exploratory study on smartphone overuse among college students. In Proceedings of the SIGCHI conference on human factors in computing systems, pages 2327–2336, 2014.

[4] Vivek K Singh and Rishav R Agarwal. Cooperative phoneotypes: exploring phone-based behavioral markers of cooperation. In Proceedings of the 2016 ACM International Joint Conference on Pervasive and Ubiquitous Computing, pages 646–657, 2016.

[5] Thomas Stütz, Thomas Kowar, Michael Kager, Martin Tiefengrabner, Markus Stuppner, Jens Blechert, Frank H Wilhelm, and Simon Ginzinger. Smartphone based stress prediction. In International Conference on User Modeling, Adaptation, and Personalization, pages 240–251. Springer, 2015.

[6] Fanglin Chen, Kewei Xia, Karan Dhabalia, and Jason I Hong. Messageontap: a suggestive interface to facilitate messaging-related tasks. In Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems, pages 1–14, 2019.
[7] Andreas Riegler and Clemens Holzmann. Ui-cat: calculating user interface complexity metrics for mobile applications. In Proceedings of the 14th International Conference on Mobile and Ubiquitous Multimedia, pages 390–394, 2015.

[8] Xiaoyi Zhang, Anne Spencer Ross, Anat Caspi, James Fogarty, and Jacob O Wobbrock. Interaction proxies for runtime repair and enhancement of mobile application accessibility. In Proceedings of the 2017 CHI conference on human factors in computing systems, pages 6024–6037, 2017.

[9] Android Developer. Android10, 2022.

[10] Apple. Apple support, 2022.

[11] Barry Brown, Moira McGregor, and Eric Laurier. Iphone in vivo: video analysis of mobile device use. In Proceedings of the SIGCHI conference on Human Factors in computing systems, pages 1031–1040, 2013.

[12] Maxi Heitmayer and Saadi Lahlou. Why are smartphones disruptive? an empirical study of smartphone use in real-life contexts. Computers in Human Behavior, 116:106637, 2021.

[13] Philipp Krieter and Andreas Breiter. Analyzing mobile application usage: generating log files from mobile screen recordings. In Proceedings of the 20th international conference on human-computer interaction with mobile devices and services, pages 1–10, 2018.

[14] Android Developer. Accessibilityservice, 2022.

[15] Android Developer. Usageservice, 2022.

[16] Denzil Ferreira, Vassilis Kostakos, and Anind K Dey. Aware: mobile context instrumentation framework. Frontiers in ICT, 2:6, 2015.

[17] Xiangming Mu. Towards effective video annotation: An approach to automatically link notes with video content. Computers & Education, 55(4):1752–1763, 2010.

[18] Daisuke Yamamoto, Tomoki Masuda, Shigeki Ohira, and Katashi Nagao. Video scene annotation based on web social activities. IEEE multimedia, 15(3):22–32, 2008.

[19] Martin Merkt, Sonja Weigand, Anke Heier, and Stephan Schwan. Learning with videos vs. learning with print: The role of interactive features. Learning and Instruction, 21(6):687–704, 2011.

[20] Chiman Kwan, Jenson Yin, and Jin Zhou. The development of a video browsing and video summary review tool. In Pattern Recognition and Tracking XXIX, volume 10649, page 1064907. International Society for Optics and Photonics, 2018.

[21] Alistair Morrison, Paul Tennent, and Matthew Chalmers. Coordinated visualisation of video and system log data. In Fourth International Conference on Coordinated & Multiple Views in Exploratory Visualization (CMV’06), pages 91–102. IEEE, 2006.

[22] Statista. Most frequently used smartphone apps in south korea as of september 2020, by category, 2020.

[23] Statista. Most used smartphone functions in south korea from 2014 to 2017, 2020.

[24] Android Developer. Notificationlistenerservice, 2022.

[25] Android Developer. Notificationmanager, 2022.

[26] Aman Kr Singh, Ashish Kr Prajapati, Vikash Kumar, and Subhankar Mishra. Usage analysis of mobile devices. Procedia computer science, 122:657–662, 2017.

[27] Choonsung Shin, Jin-Hyuk Hong, and Anind K Dey. Understanding and prediction of mobile application usage for smart phones. In Proceedings of the 2012 ACM Conference on Ubiquitous Computing, pages 173–182, 2012.

[28] Xiangang Qin, Chee-Wee Tan, Effie Lai-Chong Law, Mads Bødker, Torkil Clemmensen, Hequn Qu, and Diandi Chen. Deciphering the role of context in shaping mobile phone usage: Design recommendations for context-aware mobile services from a cross-cultural perspective. In Proceedings of the Sixth International Symposium of Chinese CHI, pages 39–48, 2018.

[29] Martin Pielot, Karen Church, and Rodrigo De Oliveira. An in-situ study of mobile phone notifications. In Proceedings of the 16th international conference on Human-computer interaction with mobile devices & services, pages 233–242, 2014.