A Systematic Survey on Android API Usage for Data-driven Analytics with Smartphones

HANSOO LEE, School of Computing, KAIST, South Korea
JOONYOUNG PARK, Graduate School of Knowledge Service Engineering, KAIST, South Korea
UICHIN LEE, School of Computing, KAIST, South Korea

Recent industrial and academic research has focused on data-driven analytics with smartphones by collecting user interaction, context, and device systems data through Application Programming Interfaces (APIs) and sensors. The Android operating system provides various APIs to collect such mobile usage and sensor data for third-party developers. Usage Statistics API (US API) and Accessibility Service API (AS API) are representative Android APIs for collecting app usage data and are used for various research purposes, as they can collect fine-grained interaction data (e.g., app usage history and user interaction type). Furthermore, other sensor APIs help to collect a user's context and device state data, along with AS/US APIs. This review investigates mobile usage and sensor data-driven research using AS/US APIs by categorizing the research purposes and the data types. In this article, the surveyed studies are classified as follows: five themes and 21 subthemes and a four-layer hierarchical data classification structure. This allows us to identify a data usage trend and derive insight into data collection according to research purposes. Several limitations and future research directions of mobile usage and sensor data-driven analytics research are discussed, including the impact of changes in the Android API versions on research, the privacy and data quality issues, and the mitigation of reproducibility risks with standardized data typology.

CCS Concepts: • **Human-centered computing** → **Ubiquitous and mobile computing**; **Mobile phones**; *Smartphones*; **Human computer interaction (HCI)**; *Interaction techniques*; **Accessibility**; • **General and reference** → **Surveys and overviews;**

Additional Key Words and Phrases: Smartphone sensing, User interaction data, data analysis, android accessibility, android API, data collection, sensor, mobile device, smartphone, data driven

ACM Reference format:
Hansoo Lee, Joonyoung Park, and Uichin Lee. 2022. A Systematic Survey on Android API Usage for Data-driven Analytics with Smartphones. *ACM Comput. Surv.* 55, 5, Article 104 (December 2022), 38 pages.
https://doi.org/10.1145/3530814

This research was supported by the Basic Science Research Program through the National Research Foundation (NRF) funded by the Korean government (MSIT) (2020R1A4A1018774, 2022R1A2C2011536).

Authors' addresses: H. Lee and U. Lee (corresponding authors), School of Computing, KAIST, 291 Daehak-ro, Yuseong-gu, Daejeon, South Korea, 34141; emails: {hansoolee, uclee}@kaist.ac.kr; J. Park, Graduate School of Knowledge Service Engineering, KAIST, 291 Daehak-ro, Yuseong-gu, Daejeon, South Korea, 34141; email: zelatore85@kaist.ac.kr.

0360-0300/2022/12-ART104 $15.00
https://doi.org/10.1145/3530814

1 INTRODUCTION

Currently, many people regard smartphones as an extension of their bodies, supporting everyday lives. According to recent statistics, 79% of adults carry their mobile phones for 22 hours a day [2], and users spend an average of 3 hours and 15 minutes a day [106]. Users interact with their smartphones an average of 2,617 times a day [160]. While holding or using a smartphone, fine-grained contextual information such as surrounding context, physical activity, and app usage patterns can be collected from built-in sensors and **application programming interfaces (API)**. Thus, smartphones provide new opportunities to analyze a user's everyday life patterns. For example, mobile usage and sensor data serve as a valuable resource for the diagnosing and prognosis of mental and physical health [86, 103].

This review focuses on the Android mobile **operating system (OS)**, which dominates 83.8% of the global smartphone market [82]. As shown in Figure 1, Android built-in APIs can identify a user's life patterns based on mobility and activity data. Thus, we can collect interaction data (e.g., each app's touch event). Android OS allows developers to customize these built-in APIs to build applications for diverse passive data collection. For this reason, Android has been actively used in the research community. Researchers commonly use the **Accessibility Service API (AS API)** to collect user interaction and app usage pattern data among Android's built-in APIs, also called Accessibility API (e.g., *AccessibilityService* and *AccessibilityEvent*) [43]. We can track app usage using various *AccessibilityEvent* classes (e.g., transition type, view type, exploration type, and notification type) [35].

Another way to collect app usage pattern data is using the Android Usage Statistics APIs **(US APIs)** [38]. US API can be used to collect the app and device usage history and statistics. US API can access device usage history and statistics to obtain the currently running app information in third-party applications (Android API level 21 in 2014). Therefore, the US API has been mainly used to collect app usage pattern data in many studies (e.g., References [63, 90, 108, 144]). Beyond simple usage tracking, the AS API is still exclusively used to track the detailed **user interface (UI)** status information (e.g., UI changed status, interaction type, UI elements and hierarchy, and notification) (e.g., References [25, 84, 92, 164]).

Along with the AS/US APIs, other Android built-in sensors and APIs are also used to collect mobile usage and sensor data for the research. These data can be collected using Android built-in **hardware (HW)** sensors (e.g., accelerometer, gyro, GPS, Wi-Fi, Bluetooth, proximity, light, compass, pressure) and APIs (e.g., Sensor, Wi-Fi, Google Location Services APIs, Bluetooth APIs). In other words, we can use these Android built-in sensors and APIs to track a user's physical activity, physiological state, and surrounding context information (e.g., location, ambient environment) for the research. Furthermore, Android system APIs (e.g., *BroadcastReceiver*, *BatteryManager*) can be used to collect device status information (e.g., battery information, screen status, device information setting, and ringer mode) (see Figure 1).

Leveraging data-driven analytics with smartphone data is widely performed in various research fields (e.g., sensors, human-computer interaction, ubiquitous computing, and mobile computing) as follows:

- Understanding a user's smartphone usage pattern [30, 84, 94].
- Identifying a user's notification checking factors, how the user is affected by the notification, and message management and scheduling [25, 123, 154].
- Understanding a user's psychological and health states and personal traits in digital phenotype and digital health research fields [1, 18, 136].
- Validating permission, improve authentication and authorization and improve protection from malware in privacy and security research fields [67, 88, 118].

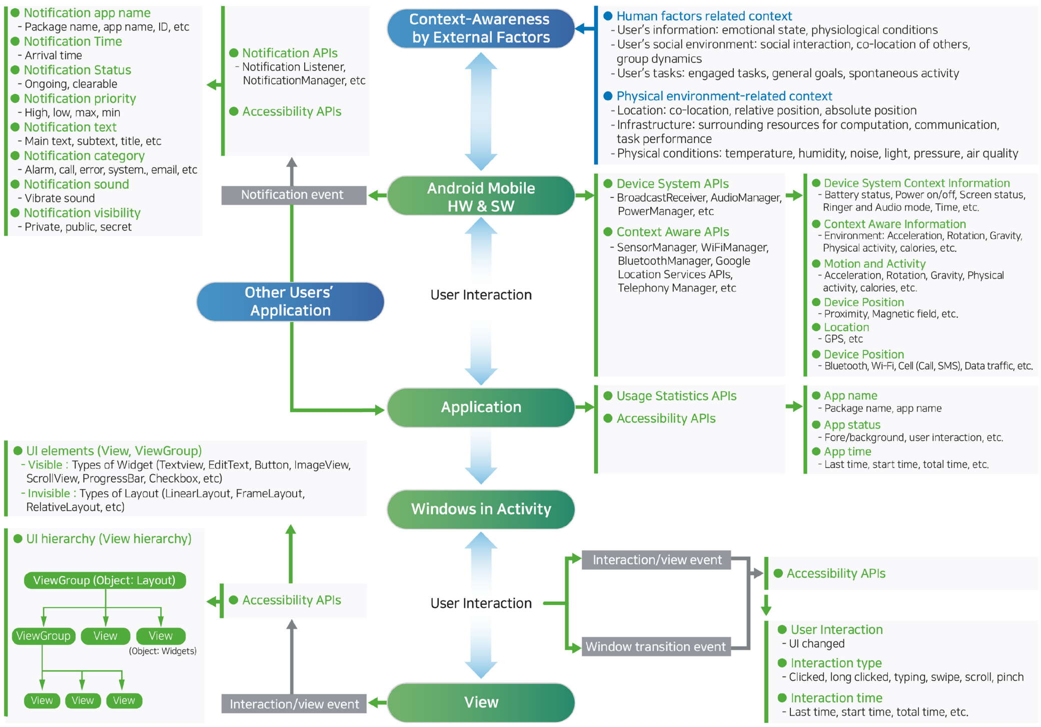
Notification app name
- Package name, app name, ID, etc
Notification Time
- Arrival time
Notification Status
- Ongoing, clearable
Notification priority
- High, low, max, min
Notification text
- Main text, subtext, title, etc
Notification category
- Alarm, call, error, system., email, etc
Notification sound
- Vibrate sound
Notification visibility
- Private, public, secret
Notification APIs
- Notification Listener, NotificationManager, etc
Accessibility APIs
Notification event
Other Users' Application
Context-Awareness by External Factors
Human factors related context
- User's information: emotional state, physiological conditions
- User's social environment: social interaction, co-location of others, group dynamics
- User's tasks: engaged tasks, general goals, spontaneous activity
Physical environment-related context
- Location: co-location, relative position, absolute position
- Infrastructure: surrounding resources for computation, communication, task performance
- Physical conditions: temperature, humidity, noise, light, pressure, air quality
Android Mobile HW & SW
User Interaction
Device System APIs
- BroadcastReceiver, AudioManager, PowerManager, etc
Context Aware APIs
- SensorManager, WiFiManager, BluetoothManager, Google Location Services APIs, Telephony Manager, etc
Device System Context Information
- Battery status, Power on/off, Screen status, Ringer and Audio mode, Time, etc.
Context Aware Information
- Environment: Acceleration, Rotation, Gravity, Physical activity, calories, etc.
Motion and Activity
- Acceleration, Rotation, Gravity, Physical activity, calories, etc.
Device Position
- Proximity, Magnetic field, etc.
Location
- GPS, etc
Device Position
- Bluetooth, Wi-Fi, Cell (Call, SMS), Data traffic, etc.
Application
Usage Statistics APIs
Accessibility APIs
App name
- Package name, app name
App status
- Fore/background, user interaction, etc.
App time
- Last time, start time, total time, etc.
UI elements (View, ViewGroup)
- Visible : Types of Widget (Textview, EditText, Button, ImageView, ScrollView, ProgressBar, Checkbox, etc)
- Invisible : Types of Layout (LinearLayout, FrameLayout, RelativeLayout, etc)
Windows in Activity
UI hierarchy (View hierarchy)
ViewGroup (Object: Layout)
ViewGroup
View
View
(Object: Widgets)
View
View
View
Accessibility APIs
User Interaction
Interaction/view event
Window transition event
Accessibility APIs
User Interaction
- UI changed
Interaction type
- Clicked, long clicked, typing, swipe, scroll, pinch
Interaction time
- Last time, start time, total time, etc.
Interaction/view event
View

Fig. 1. Mobile usage and sensor data collection through Android APIs.

- Improving user interaction, accessibility, and optimization [26, 97, 137].
- Testing and programming support fields, such as **graphical user interface (GUI)** automated testing, crowdsourced system testing, programming support for developer, and app performance measurement [14, 77, 112].
- Development of frameworks and platforms to collect smartphone data [81, 111].

Table 1 lists the existing review studies investigating prior works that used various mobile usage and sensor data for specific research purposes. Rooksby et al. [136] reviewed existing studies that used smartphone sensor data to monitor student mental health. Kourtis et al. [86] investigated the usage of sensor data by reviewing 62 studies that used mobile and wearable devices to determine digital biomarkers for Alzheimer's disease. Liang et al. [103] reviewed 92 studies that used various types of data (e.g., online activities, blood pressure, ECG, EEG), including mobile and wearable device data, in a digital phenotyping study of mental health.

There are also review studies investigating research that used various mobile usage and sensor data under different experimental conditions or focused on specific mobile usage and sensor data as depicted in Table 1. To gain a more generalizable understanding of mobile device use across different user populations, Church et al. [30] analyzed the differences in data collected in 22 papers according to the number of different experimenters, experiment period, deployment, recruitment, incentives, and methods. Berkel et al. [152] investigated 110 studies using the **experience sampling method (ESM)** in experiment duration, types of triggers, response rate, compensation, personal device, and utilized data.

However, no prior studies performed a systematic survey of the existing research papers based on Android's AS/US APIs and other built-in sensing APIs to the best of our knowledge. In addition,

Table 1. Papers Including a Summary of Papers Using Mobile Usage and Sensor Data

Ref.	Number of Reviewed Papers	Reviewed Items	Type of Investigated Data	Limitation
Church et al. [30]	22	Number of participants, experiment duration, deployment, recruitment, incentives, method, data collected, user input	Time, communication, mobility, battery, screen, notifications, multimedia, applications, locations, touch	For the 22 reviewed papers, only the experimental design method (e.g., number of participants, period, recruitment, compensation) was analyzed, and there is no classification of the research purpose, so it is difficult to understand in detail what data was used for what research purpose.
Rooksby et al. [136]	19	Utilized data, experiment duration, number of participants	Acceleration, activity, app usage, battery, charge, browser history, Bluetooth, call logs, camera events, creen, keyboard, UI, location, light sensor, microphone, SMS/email	In 19 studies, what kind of sensor data was used was investigated, but only the research purpose o/f a limited field (i.e., student mental health monitoring in digital phenotyping) was investigated.
Kortis et al. [86]	62	Utilized data, sense domain, metrics	Camera, microphone, acceleration/gyro, barometer, touchscreen, geoposition, device use, ECG, PPG, IR thermometer, ballistocardiography, galvanic skin response, ambient light sensor, UV sensor, electromyogram (EMG)	This research investigated 62 papers using mobile/wearable sensor data only for specific research purposes (i.e., Alzheimer's digital biomarker). Furthermore, none of the 62 papers collected app usage data.
Berkel et al. [152]	10	Experiment duration, trigger, response rate, compensate, personal device, utilized data	ESM, location, phone calls, network related, bio-signal (s), accelerometer, app usage, other (s)	The 110 papers using a specific method (experience sampling method on a mobile device) were reviewed only, and the data used in each paper was briefly presented without data categorization as the name of the device's sensor.
Liang et al. [103]	92	• Data collection strategies (data, strategy, pros, cons) • List of affect lexicons (category, lexicon, affect the type, vocabulary) • Applications for mental health (type, name, symptom, data) • Affect recognition (modality, category, work, algorithm) • Behavior anomaly detection (category, sensor, work, algorithm)	• Physical data: facial expressions, behavioral data, vocal data, context data (e.g., location, timestamp) • Cyber data: online behaviors, social media data • Biological data: blood oxygen saturation, blood pressure, heartbeat rate, respiration frequency, brain waves • Social data: call logs, text messages, GPS location, hot spots, email, friendship list, online activities (e.g., like, comment, sharing)	Only 92 studies were reviewed for limited research purposes (i.e., digital phenotyping of mental health), and the review did not focus on papers using various mobile sensor and usage data.

the existing review papers do not give a holistic view of mobile usage and sensor data types for research purposes, because they only investigated particular content (i.e., specific research purposes, data type, and experimental conditions). Furthermore, mobile usage and sensor data terminology are named differently, making it challenging to correctly identify mobile usage and sensor data items.

This work first identifies the primary research purposes of those studies that use AS/US APIs. Then, this research reviewed the types of mobile usage and sensor data according to research purposes. Last, this research systematically reviewed the existing studies and categorized research purposes via a thematic analysis with affinity diagramming. In addition, this research categorized the collected data as follows: i.e., physical context sensing, system sensing, and interactive sensing data, by referring to API documents and prior studies.

The main contributions of this work can be summarized as follows:

- This study provides an overview of the trends in existing research over the past 10 years by categorizing the reviewed papers into several research themes through thematic analysis.
- Next, this study reports on how to collect and use the data and summarizes the used data types according to surveyed studies and research purposes.
- Our findings help us to identify the relationships between research purposes and data types in mobile usage and sensor data-driven analytics research.
- This study discusses challenges and insights (e.g., privacy and data quality issues, reproducibility issues of data term confusion, and research trends according to APIs release and update) that may arise when conducting various mobile usage and sensor data-driven analytics research.

The rest of this article is organized as follows. Section 2 provides background information on AS/US API, built-in sensors, and the other APIs. In Section 3, we present the overall process: finding, classifying, and selecting the literature covered for the systematic survey. In Section 4, we classify the research purpose in surveyed studies. In Section 5, we categorize the mobile usage and sensor data. In Section 6, we present the results of data categorization. In Section 7, we summarize the key findings from classification results; discuss the open issues (i.e., data term issues and practical issues in AS/US API use), privacy and data quality issues, influence of API updates on research; and list the limitations of this study and scope for future research. Finally, we conclude this article in Section 8.

Table 2. Macro Constants for Service Request According to Android Manager Type [47]

Constants: Requested Service	Managers: Returned Object	Description of Managers
ACCESSIBILITY_SERVICE	AccessibilityManager	Provides user feedback on UI events through registered event listeners.
USAGE_STATS_SERVICE	UsageStatsManager	Provides to the information of app and system usage history and statistics.
NOTIFICATION_SERVICE	NotificationManager	Access to information for the notification events to users that happen in the system and app.
SENSOR_SERVICE	SensorManager	Access to the sensors related information in Android device.
TELEPHONY_SERVICE	TelephonyManager	Access to information about telephone communication services.
LOCATION_SERVICE	LocationManager	Access to information about location based services (LBS).
AUDIO_SERVICE	AudioManager	Access to information about volume and ringer mode control.
BLUETOOTH_SERVICE	BluetoothManager	Obtain the BluetoothAdapter resources and manage overall Bluetooth.
WIFI_SERVICE	WIFIMananager	Manage the overall Wi-Fi network information.

2 BACKGROUND

This section describes Android built-in APIs used to collect mobile usage and sensor data described in Figure 1. Section 2.1 explains the Android OS architecture and basic API framework (i.e., manager) for collecting mobile usage and sensor data. Section 2.2 illustrates the APIs used to collect application interaction data such as user interaction and app usage patterns. Section 2.3 explains the APIs used to collect context-aware data (location, physical activity, ambient environment, network). Section 2.4 describes the APIs that can track device and system status (e.g., battery and power status, screen on/off). Section 2.5 demonstrates why this study focuses on research using AS/US APIs among Android built-in APIs.

2.1 Background on Android APIs to Collect Mobile Usage and Sensor Data

This research describes the essential components and API framework to understand the Android APIs for mobile usage and sensor data collection. The Android OS consists of four layers that are the application layer (system apps and third-party apps), Framework layer (API framework), Middleware layer (Native C/C++ library, Android runtime, Hardware Abstraction Layer), and kernel layer (Linux kernel) [57, 109]. Third-party apps can access and utilize mobile usage and sensor data through the Android API framework's manager objects such as *AccessibilityManager*, *UsageStatsManager*, *NotificationManager*, and *SensorManager*, which are system-level services provided by Android. An instance of each object can be obtained through the *getSystemService()* function [47], and specific data can be accessed via each instance. Table 2 shows constant parameter values in the process of obtaining an instance of each object and describes the Android API framework's manager objects that can be accessed through each instance.

Furthermore, there is an alternative way to collect mobile usage and sensor data besides accessing the manager in third-party apps. First, BroadcastReceiver can detect system status events and information (e.g., screen status, power status, and battery status) occurring in Android OS. In addition, Bluetooth status information can be collected using only Bluetooth APIs, such as *BluetoothAdapter* and *BluetoothDevice*, without going through *BluetoothManager* [44]. To collect location information, Google recommends that using the Google Location Services APIs is a more straightforward method for higher accuracy than the Location APIs that provide existing Android location-based and related services such as *LocationManager* and *LocationProvider* [42]. In some cases of UI event collection or third-party app development through AS API, *AccessibilityService*, and *AccessibilityEvent* are only used except for *AccessabilityManager* (detailed explanation in Section 2.2). Through *UsageStatsManager*, the researchers can collect usage history and statistics of apps and systems. Accordingly, *UsageStatsManager* and *AccessibilityService* are used more than other Managers and Services to identify smartphone usage patterns. First, the AS/US API can collect app usage and interaction, which are the core aspects of this review in Section 2.2. Second, context-awareness APIs (i.e., collecting ambient environments of the smartphone) and device and system status APIs (i.e., collecting internal system and device status) that can be used in the study along with AS/US APIs are described in Sections 2.3 and 2.4, respectively.

2.2 Application Interaction Data Collection APIs

As shown in Figure 1, we present an overview of the main functions, history, and collectible data of AS/US APIs. This overview aims to clarify the Android APIs used for app usage pattern data collection and derive the appropriate API terms for the literature search terms.

2.2.1 *Accessibility Service APIs.*

Accessibility APIs are a set of Android APIs that are the basis for building applications to enhance user interfaces to support users with disabilities or users who are temporarily unable to fully interact with a device. Accessibility APIs include AccessibilityService and AccessibilityEvent. In this study, Android developer's documentation says that "the ability for you to build and deploy accessibility services was introduced with Android 1.6 (API Level 4) and received significant improvements with Android 4.0 (API Level 14)" [48]. Further, "Accessibility services should only be used to assist users with disabilities (e.g., visually impaired people) in using Android devices and apps" as per Android developer's documentation [35]. Besides, Google Android provides an application made with standard AS API (e.g., TalkBack), an assistive technology with functions such as screen touch, voice, Braille-based screen reader, speaker-based voice, and vibration [76]. For research purposes, AS API has been widely used to collect fine-grained data related to the user interaction data (e.g., types of interactions and app usage status), which provides useful information for behavior analysis.

The collection of fine-grained data (i.e., UI information via user interaction) is possible by grasping an *AccessibilityEvent* delivered to the AS API through the *onAccessibilityEvent()* callback method when the UI status is changed. The *AccessibilityEvent* mainly appears when an event occurs in the UI due to touch manipulation, notification, or system change. First, to use the AS API, the service element should be included in the application element of the manifest file. The intent filter of the AS API must also be included in the service element. Next, the BIND_ACCESSIBILITY_SERVICE permission must be specified. The AS API can inform the system how and when to execute the AS API through a configuration variable set. For example, the developer can designate the *AccessibilityEvent* type and package name to be processed by the AS API. Therefore, when UI events occur in the Android system and app, the fine-grained data can be collected through AS API as follows: Types of touch and gesture interaction (e.g., click, long click, scroll, typing), time of interaction (e.g., interaction start/end time), view hierarchy construct, view elements, notification state change, and windows state change. As shown in Table A3 in Appendix C, the types of *AccessibilityEvent* are categorized mainly into VIEW TYPES, TRANSITION TYPES, NOTIFICATION TYPES, EXPLORATION TYPES, and MISCELLANEOUS TYPES, for example, in Android 11 (API level 30), and there are currently a total 47 types of *AccessibilityEvent* [34].

2.2.2 *Usage Statistics APIs.*

US API is provided mainly to obtain device and application usage history and statistics information [38]. The US API includes the representative APIs (e.g., *UsageStatsManager*, *UsageEvents*, *StorageStatsManager*, *ConfigurationStats*, and *EventStats*) for accessing the app, as well as device, network, storage, device configuration, event type usage history, and statistic [38]. US API has been widely used to analyze the application and device usage patterns along with AS API in many existing studies. Prior to Android 5.0 (API level 21), *ActivityManager* was used to obtain information about a currently running foreground application. However, as the method of getting the currently running application information using *ActivityManager*'s methods (e.g., *getRunningTask()*, *getRecentTasks()*, and *getRunningServices()*) have been deprecated since Android 5.0 (API level 21) [36]. Instead of *ActivityManger*, US API is used mainly to track device and application usage history and statistics information. Among the many classes in the US API above, we focus on the classes that can access app/device usage history and statistics such as *UsageStatsManager*, *UsageEvents*, and *UsageEvents.Event* [10–12].

UsageStatsManager can query application records. For example, by using the usage statistics constants (i.e., INTERVAL DAILY, INTERVAL WEEKLY, INTERVAL MONTHLY, and INTERVAL YEARLY), we can query the records to receive the classified results according to different periods (i.e., by the day, the week, the month, and the year) [12]. Moreover, app usage time (e.g., last time used, app foreground/background used time), app usage status history (e.g., the status of foreground/background/user interaction), and package name can be retrieved using the *UsageStats* object provided in the *UsageEvents.Event* class through methods such as *getPackageName()*, *getLastTimeUsed()*, and *getTotalTimeInForegroud()*. Furthermore, five types of bucket information (active, working set, frequent, rare, never) that define the usage status for each app according to how often the app was used recently can be obtained through *UsageStatsManager* as of Android 10 (API level 29) [12]. Various Android runtime information can be collected along with various application pattern information in US API. In addition, we can check the detailed information collected through *UsageStatsManager* [12] and *UsageEvents.Event* [11] in Table A4 in Appendix C.

2.2.3 *Notification API Framework.*

Notification APIs refer to all APIs related to the collection of notification information that occurs by app usage or device status change in the smartphone. Representative Notification APIs include AS API, *NotificationManager*, and *NotificationListenerService*. *AccessibilityEvent* in AS API has TYPE NOTIFICATION STATE CHANGED, which captures the notification information (e.g., time, app package name, displayed texts of the toasts such as small popup, and text information) notified through *getEventType()*, *getPackageName()*, *getEventTime()*, *getParcelableData()*, and *getText()* methods. However, there is a limit to the information that can be accessed through AS API. *NotificationListenerService* (launched Android API level 21) and *NotificationManager* (launched Android API level 1) can collect the more specific notification information (e.g., notification app name, time, status, priority, text, category, sound, visibility). Currently, notification-related Android APIs such as *NotificationListenerService* and *NotificationManager* are widely used along with AS API to study notification usage behaviors in existing studies (e.g., References [24, 84, 90, 91, 108, 124]).

2.2.4 *Call and Message APIs.*

Call and Message API refers to APIs related to collecting app usage history/status of call and message (e.g., **short message service (SMS)**, **multimedia message service (MMS)**), and sending/receiving notification information. Representative Call and Message APIs include AS/US API, *TelephonyManager*, *PhoneStateListener*, *SmsManager*, and *CallLog*. Existing studies have collected the call-, SMS-, and MMS-related app usage history/status, and sending/receiving notification information through AS/US API (e.g., References [5, 6, 25, 61, 94, 123, 140, 141, 159]). However, AS/US API can access only the call/message-related app usage history and status information. Therefore, telephony APIs (e.g., *TelephonyManager*, *PhoneStateListener*, and *SmsManager*) or *CallLog* can be alternatively used to access fine-grained information. *TelephonyManager* can check the access and status of information about the device's telephony service (e.g., SMS, MMS, and call) [59]. In addition, we can use the methods of *TelephonyManager* to check telephony services/status, access some types of subscriber information, and register listeners to receive phone status change notifications. Further, *CallLog* can collect call-related data such as incoming/outcoming history information [46].

2.3 Context-awareness Android APIs

Android APIs are mainly used for collecting context data as follows: motion, position, environmental context APIs, location context APIs, and network context APIs.

2.3.1 Motion, Position, and Ambient Environment Context APIs. Representative Android APIs used to collect motion, position, and ambient environment context information include HW sensor-based sensor APIs such as *SensorManager*, *Sensor*, *SensorEvent*, and *SensorEventListener* [41]. Hardware sensors are specifically classified into motion detection sensors (e.g., accelerometers, magnetometers, and gyroscope), physical position sensors (e.g., accelerometers, magnetometer, and proximity), and ambient environment sensors (e.g., light, pressure, humidity, and temperature) [58]. The context information based on mobile sensors and sensor APIs is collected and utilized in many existing studies (e.g., References [24, 61, 72, 81, 120, 124]).

Besides, the *Google Activity Recognition API* identifies the activity (i.e., walking, running, driving, standing still, and cycling) is being performed by the user at each time with the sensors in the smartphone device through detecting a user's specific activity type constants (i.e., IN_VEHICLE, ON_BYCYLE, ON_FOOT, RUNNING, STILL, TILTING, WALKING) [75]. Furthermore, the *Google Activity Recognition Transition API* can detect a user's specific activity type constants (i.e., IN_VEHICLE, ON_BYCYLE, RUNNING, STILL, WALKING) to identify when a user starts or stops a specific activity [49]. Existing studies used mobile sensors and Google Activity Recognition APIs to analyze the physical activity states (e.g., References [5, 25, 140, 141]).

2.3.2 Location Context APIs. Location APIs (e.g., *LocationManager* and *LocationProvider*) and *Google Location Services API* are representative API frameworks mainly used to collect location context information in many existing studies (e.g., References [30, 84, 159]). In terms of efficiency and accuracy, *Google Location Services APIs* are superior to the Location APIs, which can access the location information such as timestamp, latitude, longitude, altitude, accuracy, and speed of location [8]. If the *Google Location Services API* is not available, then the location information can be collected via the Location APIs (e.g., *LocationProvider* and *LocationManager*) in the traditional way as follows [42]: (1) GPS location provider (GPS, AGPS), network location provider (GPS, Cell ID, Wi-Fi MAC ID), and the authority (android.permission.ACCESS_FINE_LOCATION or android.permission.ACCESS_COARSE_LOCATION) receive information from the network location provider (GPS, Cell ID, Wi-Fi MAC ID) and passive location provider (Cell ID, Wi-Fi MAC ID) are set in the manifest file [52]. (2) After setting the authority, set the provider to be used by *LocationManager* and get updated location information from the GPS and network through the *requestLocationUpdates()* method in the *LocationManager*. It is also possible for a third-party app to collect the location provider and information (e.g., from the *LocationManager* in real-time through *LocationListener*) [51].

2.3.3 Personal and Local Area Network Context APIs. Smartphone network context information can be primarily divided into **personal area network (PAN)** and **local area network (LAN)** information. Android's built-in Bluetooth sensor and APIs (including **Bluetooth low energy (BLE)**) such as *BluetoothManager*, *BluetoothAdapter*, and *BluetoothDevice* are mainly used to collect PAN usage data (e.g., device name, type, address, and connection status) [39]. BLE scanning such as *BluetoothLeScanner* [40] can be used to search for nearby BLE devices. Wireless LAN information can be collected using Android built-in network sensors (e.g., Wi-Fi) and network APIs such as *WifiManager* and *ConnectivityManager*; e.g., **Service Set Identifier (SSID)**, **Basic Service Set Identifiers (BSSID)**, Received Signal Strength Indicator, frequency, and the presence of connectable Wi-Fi networks can be detected in real-time (known as Wi-Fi scanning). *ConnectivityManager* can be utilized to collect the network connection state information. In addition, *NetworkStatsManager* and *TrafficStats* can be used to obtain the network traffic usage statistics and history information (e.g., transmitted and received information of the network packets and bytes from all interfaces and mobile) [9, 54].

2.4 Device and System Status APIs

Android devise and system status information can be obtained by tracking events that occur according to real-time operation states (e.g., battery states, power on/off, network connection states) using the device and system status APIs. Currently, there are two main methods used to obtain device and system status information in Android OS. The first method is to register *BroadcastReceiver* and selectively access device/system events using *IntentFilter* to receive device/system status information and resources in Android [45]. When a specific device system status change occurs in the Android system, event information (e.g., ACTION_AIRPLANE_MODE_CHANGED, ACTION_BATTERY_CHANGED/LOW/OKAY, and ACTION_BOOT_COMPLETED) is delivered in the form of intent through the *BroadcastReceiver* [50]. At this time, the third-party app can access the detailed device system information (e.g., battery statuses change, battery charge statuses such as low, high, system booting completion, bug reporting, phone connection, call button press, date change, system reboot, and device system status information). The second method is to access the manager (e.g., *BatteryManager*, *AlarmManager*, *PowerManager*, and *AudioManager*) related to the device system by using the *getSystemService()* method. For example, the *BatteryManager* can broadcast all battery and charge details to a sticky Intent containing the state of charge without registering the *BroadcastReceiver* [53]. Through *BatteryManager*, third-party apps can access the device battery states information such as battery health (cold, dead, good, overheat, over-voltage), battery plugged (AC, USB, wireless), battery property (e.g., capacity, charge counter), and battery status (charging, discharging, full, not charging).

2.5 Discussion of Android APIs

This section dealt with Android APIs for collecting mobile usage and sensor data. To track various mobile usage and sensor data, the types of Android APIs were classified into three categories, and the data that each type of Android API can collect were investigated. Among the Android APIs described above, AS/US APIs are the most representative APIs used in smartphone usage research targeting human behavior tracking by logging app usage, touch interactions, keystrokes, and notifications. Furthermore, most of the data collected by the context-aware Android APIs in Section 2.3 and device and status APIs in Section 2.4 can be collected through AS/US APIs as well. Therefore, recent studies tended to collect app usage logs corresponding to user interaction, notification, and device and system status, through AS/US APIs. Context-awareness APIs and device and system status APIs were additionally used for further data collection, such as ambient environment (e.g., location, temperature, and humidity), physical activity state, network state (e.g., Bluetooths and Wi-Fi status, data traffic), and detailed device and system state changes (e.g., battery status, screen on/off, power on/off). For this reason, this research intends to focus on the studies that used AS/US API among various Android APIs.

3 METHODOLOGY

3.1 Literature Search

This study created a keyword list and searched using those terms to find studies in which AS/US APIs were used on four scholarly literature search engines: Google Scholar,[1] Web of Science,[2] Scopus,[3] ScienceDirect,[4] and ACM Digital Library.[5] Because AS/US APIs are used in various

[1] https://scholar.google.co.kr/.
[2] http://www.webofknowledge.com/.
[3] https://www.scopus.com/.
[4] https://www.sciencedirect.com/.
[5] https://dl.acm.org/.

Table 3. List of Android, Accessibility APIs, Usage Statistics API, and Combined Keywords

List	Keywords
Android-related	"android"

Accessibility APIs-related	"accessibility"; "accessibility service"; "accessibilityservice";
	"accessibility API"; "accessibility framework";
	"accessibility event"; "accessibilityevent"

Usage Statistic API-related	"usagestats"; "usagestatsmanager"; "usageevent"

Combined	"android" and "accessibility"; "android" and "accessibility service";
	"android" and "accessibilityservice"; "android" and "accessibilityevent";
	"android" and "accessibility event"; "android" and "accessibility API";
	"android" and "accessibility framework"; "android" and "usagestats"

research fields, we searched for papers in the popular literature search engines [78, 107, 149] as in prior review studies, in which papers from different fields can be found instead of the web library databases of specific domains.

The keywords were divided to search papers into four lists: Android related, Accessibility APIs related, Usage Statistic API related, and combined keywords. This keyword list was composed of terms that are frequently used in the existing literature. Table 3 presents the list of the keywords used to search the papers. AS API became available in Android 1.6 (API level 4) in September 2009, and US API became available in the Android Framework from Android 5.0 (API level 21) in October 2014. Therefore, published papers were searched using AS/US APIs for the period from 2009 to 2020. The literature search keywords, period, language, and sites are summarized in Figure 2.

3.2 Literature Filtering and Classification

The studies were classified into two main categories: The studies were classified into two main categories: (1) studies on the application or framework development for collecting fine-grained datasets and (2) studies on collecting fine-grained datasets using the AS/US API. As shown in Figure 2, we used the customized **Preferred Reporting Items for Systematic Reviews and Meta-Analyses (PRISMA)** flow diagram [110] for our review to visualize the filtering and classification process for finding target papers. In addition, the classification results according to the four filtering criteria were divided into four categories of **included papers (IP)** to be reviewed and four categories of excluded papers to be excluded in our survey scope.

Filtering criteria 1 studies corresponding to journals or proceedings, and filtering criteria 2 studies that mention the Android AS API or US API in the mobile environment. For filtering criteria 3 studies, we distinguished the existing studies using the AS to improve the interface accessibility of the users with disabilities (the original purpose of the Google Android policy) from papers that focused on AS API for other research purposes. Filtering criteria 4 studies used the AS or US API for generic research purposes beyond improving accessibility. According to four filtering criteria, only papers from academic journals or conferences written in English were included in the search results, and among them, papers using Android AS API or US API were selected.

The studies classified as IP1 and IP2 were considered to lie within the scope of the review. Through a review of these studies, we identified the research purpose of the studies, the types of AS/US APIs data used, and smartphone data collected via using other Android APIs. Table 4 shows the description and number of studies according to types of IP (IP1-1-AS, IP1-1-US, IP1-2-AS, IP1-2-US, IP2-1, IP2-2), which are the studies included in the scope of review. A total of 109 studies were investigated, as two studies [20, 137] were counted twice.

4 RESEARCH PURPOSE CATEGORIZATION

We categorized the major research purposes of the included studies as IP1 (i.e., studies that used AS or US API *other than* accessibility enhancement purposes for people with disabilities) in Section 4.1

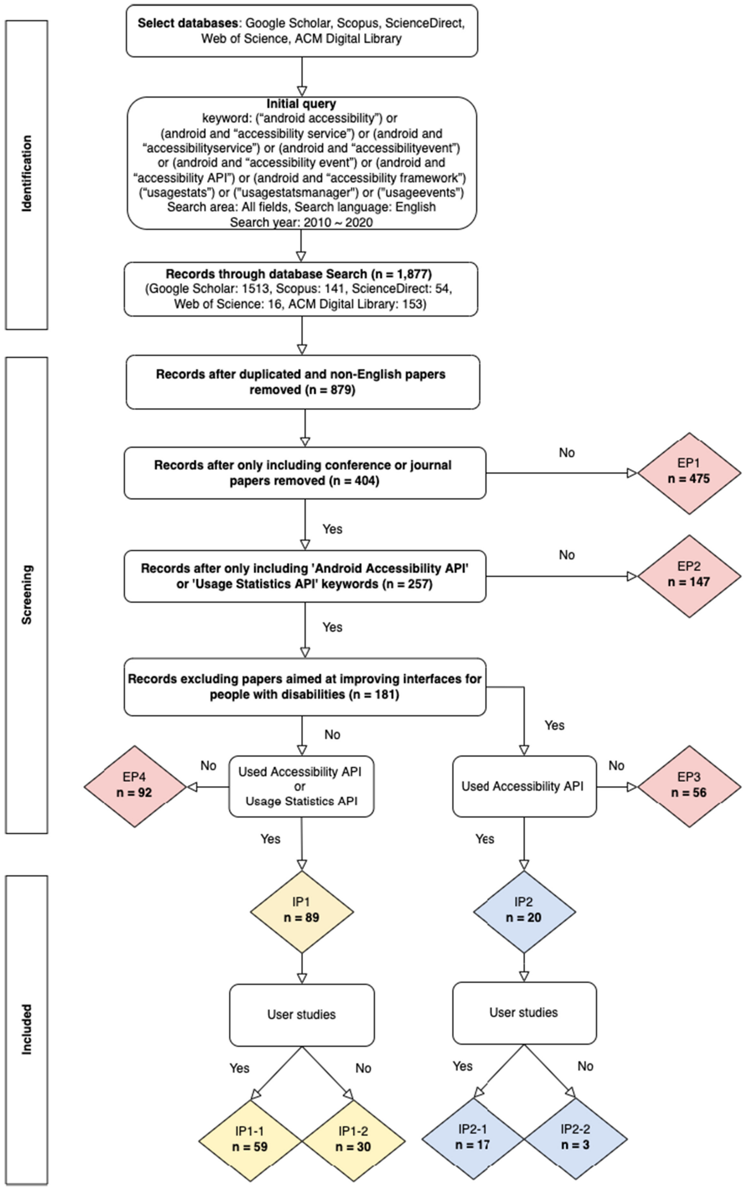
Identification
Select databases: Google Scholar, Scopus, ScienceDirect, Web of Science, ACM Digital Library
Initial query
keyword: ("android accessibility") or (android and "accessibility service") or (android and "accessibilityservice") or (android and "accessibilityevent") or (android and "accessibility event") or (android and "accessibility API") or (android and "accessibility framework") ("usagestats") or ("usagestatsmanager") or ("usageevents")
Search area: All fields, Search language: English
Search year: 2010 ~ 2020
Records through database Search (n = 1,877)
(Google Scholar: 1513, Scopus: 141, ScienceDirect: 54, Web of Science: 16, ACM Digital Library: 153)
Screening
Records after duplicated and non-English papers removed (n = 879)
Records after only including conference or journal papers removed (n = 404)
No
EP1 n = 475
Yes
Records after only including 'Android Accessibility API' or 'Usage Statistics API' keywords (n = 257)
No
EP2 n = 147
Yes
Records excluding papers aimed at improving interfaces for people with disabilities (n = 181)
No
Yes
Used Accessibility API or Usage Statistics API
No
EP4 n = 92
Used Accessibility API
No
EP3 n = 56
Yes
Yes
Included
IP1 n = 89
IP2 n = 20
User studies
User studies
Yes
No
Yes
No
IP1-1 n = 59
IP1-2 n = 30
IP2-1 n = 17
IP2-2 n = 3

Fig. 2. Flow diagram for filtering and classification of target papers based on PRISMA.

and IP2 (i.e., studies that used AS or US API of accessibility enhancement purposes for people with disabilities) in Section 4.2 (see Table 4 for the detailed types of included paper). For categorization, we used affinity diagramming, a grouping technique that discovers meaningful rules among numerous data for classification according to the research purpose [80]. The process of categorizing the research objectives of several papers using the affinity diagramming technique is as follows. First, we extracted the words of related research purposes in the order of title, abstract, keyword, main text, and published site name of each paper. Among the words extracted from the reviewed

Table 4. Descriptions of Meaning of IP

Types of IP	Descriptions	Number of Studies
IP1-1-AS	Research that perform user studies on AS API for improving the interface for the people with disability	43*
IP1-1-US	Research that perform user studies on US API	17*
IP1-2-AS	Research that did not perform user studies to improve the interface for the people with disability among the studies using AS API	28*
IP1-2-US	Research that did not perform user study among studies using US API	3*
IP2-1	Among the studies using the AS API, the studies where user studies were conducted for the purpose of research to improve the interface for the people with disability	17
IP2-2	Among the studies using the AS API, the studies where user study was not conducted for the purpose of research to improve the interface for the people with disability	3
Overall		111-2* =109

*Bonné et al. [17] and Ryu et al. [131], which used both AS/US APIs, were included in IP1-1-AS/US and IP1-2-AS/US twice, respectively. Therefore, the number of surveyed studies in this paper is 109, excluding the two of a total of 111.

papers, highly relevant words were grouped, and the words that were duplicated or unrelated to the purpose of the research were removed. Then, the research purposes representing the characteristics of the grouped words were found. The groups were then adjusted (e.g., groups of the same concept were merged, large groups were divided, small groups were merged with other groups) and named according to the research theme, and the themes were then divided into sub-themes. While reviewing the completed diagram from the beginning, we adjusted the themes and their labels.

4.1 Categorization of IP1: Generic Research Work

We identify the research purposes of IP1 studies (a total of 89 papers for IP1), which were classified into five themes via affinity diagramming: (1) **usage pattern (UP)**, (2) **notification (NT)**, (3) enhancement of user interaction and experience, (4) **privacy and security (PS)**, and (5) **programming and testing support (PTS)**. Additionally, the five major themes were classified into 18 sub-themes, as shown in Table 5.

4.2 Categorization of IP2: Research Work for People with Disabilities

We identify and classify the research objectives of the IP2 studies that use the AS API for the original purpose (i.e., to improve the user interface to support the users with disabilities) [35]. IP2 can be included in the **enhancement of UI and UX (EUI/UX)** among the themes of the research purpose of surveyed studies corresponding to IP1. However, in the case of IP2, the sub-themes were classified differently than the IP1 sub-themes. Accordingly, a total of 20 papers using the AS API for its original purpose have been primarily classified into three sub-themes.

Improvement of the Functionality of Accessibility Service for the users with Disabilities (IFASD): IFASD studies were to improve the functions of the accessibility aspect of the existing AS API. They are classified further into two sub-research purposes. The first is to develop the accessibility service with improved functions in terms of accessibility enhancement by customized AS API [48]. The second is to develop improved functions to enhance the accessibility for users with disabilities by using the AS API and physical devices (e.g., braille, external sensor) together. In addition, the studies can be divided into three groups depending on the evaluation methods: (1) evaluating through user study by comparing and analyzing the existing AS API and the newly developed AS API in the paper, (2) evaluating only the developed AS API through the user study, and (3) not evaluating through the user study. Table 6 summarizes studies in IFASD: what is proposed, what system or app is implemented compared to the basic Android AS API in terms of accessibility enhancement, and how usability tests were performed.

(Development or Evaluation of an application using Accessibility Service for the users with Disabilities (DEASD)): DEASD studies were conducted for the purpose of developing an

Table 5. Research theme and sub-theme in IP1

Theme	Sub-Theme	Description
Usage Pattern (UP)	Generic Usage Pattern Analysis (GUPA)	Research on the analysis of generic mobile usage patterns based on mobile data-driven analytics according to specific conditions (e.g., society, culture, education, health status, age group) of users
	Machine Learning-based Model and System (MLMS)	Research on machine learning models and system development to understand user characteristics based on the mobile usage pattern
	Framework and System Development (FSD)	Research on developing the framework and the system using AS/US APIs, and other Android APIs for enabling fine-grained mobile data collection
Notification (NT)	Identifying Factors responding to Notifications (IFN)	Research to understand or predict how users react to and interact with which notifications in what context
	Notification Timing and Notification Management (NT/NM)	Screening research to reduce interruption by notification, just in time notification scheduling research, and research on notification management according to the importance
	How Notification Affects a User (HNAU)	Research to identify how notifications affect users (e.g., attentiveness, emotion, action, responsiveness, interruptible, learning)
Enhancement of UI and UX (EUI/UX)	Computational Enhancement (CE)	Research for user interaction and experience improvement (e.g., upgrading system performance, lip recognition) through computational enhancement (e.g., automation, recognition, offloading, microservice)
	User Interface Enhancement (UIE)	Research for improving UI (e.g., personalized conversation interface, conversation-aware interface, verification by password-less access, multi-modal interface)
	User Interface Profiling (UIP)	Research on UI profiling such as complexity measurement of the UI or power management of CPU/GPU
Privacy and Security (PS)	Authentication System/Scheme (ASS)	Research on the suggestion of systems/schemes for user authentication in the Android mobile environment
	Access and Permission Control (APC)	Research to identify user intention and context for user access and permission control or to suggest new methods
	Privacy-Preserving System (PPS)	Research to propose a system for preserving user privacy in Android mobile environment
	Monitoring and Detection for Privacy and Security (MDPS)	Research to detect and monitor spying behavior and hidden attacks in terms of personal privacy and security protection
Programming and Testing Support (PTS)	GUI Automated Testing (GAT)	Research to propose an automated model and system for GUI testing of Android applications
	Crowdsourced Testing (CT)	Research to test Android applications through crowdsourcing
	Automated Programming Support (APS)	Research on supporting the automation of programming tasks in system development
	Performance Measurement and Management (PMM)	Research to measure or manage the performance of Android devices and systems
	Test Recorder and Script Generator Technique (TRSGT)	Research on technology to automatically generate test scripts by converting and encoding test actions into test script format via user interactions for automated testing

application program (e.g., calculator, configurable game, three-dimensional (3D) printing) using the AS API rather than improving the limitations of the existing AS API. The research purpose of these papers is further divided into two subcategories. The first sub-theme is to enable TalkBack for verification of interactions with developed applications. In the second sub research purpose, AS API, such as TalkBack, is enabled and used not only for interaction purposes for the developed application programs but also for application development as its own AS API. Table 6 summarizes studies in DEASD: what the research purpose is, what functions of the AS API were used, and how usability tests were performed about studies in DEASD.

Comparison of Functions of Accessibility Service for the users with Disabilities: In **Comparison of Functions of Accessibility Service for the users with Disabilities (CFASD)** studies, AS API (e.g., TalkBack) is used to find the best interaction method for users with disabilities. The ultimate goal is to enhance smartphone use for users with disabilities. Most studies were conducted to solve the accessibility issues in smartphone utilization for visually impaired people. By contrast, Zhong

Table 6. Analysis of Research in IP2: (1) What Is Proposed, (2) Accessibility Service Function Used or Improved, and (3) Usability Test Method According to the Research Purpose (IFASD, DEASD, and CFASD)

Theme	Ref.	What is Proposed	Implemented Functions or Utilized Functions in Accessibility Service	Methods of Usability Tests		
				Number of Participants	Disabilities of Participants	Evaluation Metrics
IFASD	Chinchole et al. [29]	A low-cost auxiliary development device with economical efficiency, high efficiency, mobility, and ease of use via AI and detection sensors	AI-based image recognition, collision detection, and obstacle detection functions to navigate the environment	Uncounted	Non-disabled	–
	Oliveira et al. [33]	An improvement of voice command to improve the interaction for visually impaired people	Voice commands improvement and used Talkback to evaluation	6	Visually impaired	System Usability Scale [21], setup time duration, qualitative questions autonomy, satisfaction, daily use, fatigue, and feeling of being lost
	Rodriques et al. [133]	Macros to enhance time spent searching and inaccessible content with visually impaired	Manual macro creation, automatic macro generation, and macro production to perform multiple steps of interaction with a single tab	1	Visually impaired, speech disorder, motor impaired	Interviewed user experience
	Rodriques et al. [134]	A tutorial guide for the Accessibility Service	Interactive non-visual tutorials that support audio recording and launch tutorials	11	Visually impaired	Success rate of tasks
	Trindade et al. [151]	A Hybrid-Brailler that integrate physical and touch screen devices to provide fast, accurate, and flexible nonvisual input for visually impaired users	Input solution with combining physical Braille keyboard and touch screen devices	11	Visually impaired	Words per min, typing error rate, editing ratio, task completion time
	Zhang et al. [166]	Interaction proxies to runtime repair and accessibility improvement of app.	Interaction proxies	14	Low vision, visually impaired	Accessibility barriers, user experience, usefulness and potential of proposed method based on interview
	Zhang et al. [167]	Robust annotations for app interface element to solve the access problem of the users with disabilities in Accessibility Service	Capture tool, template screen tool, runtime library, alternative collection and annotation tools to get properties of each element from Accessibility Node	5, 1	Developer, visually impaired	- Screen equivalence heuristics: the error rate of misjudgment of two screens - Runtime repair based interview
	Zhong et al. [169]	A voice control to use with other Android apps and talkback to support the system quickly	Hands-free voice control based on Accessibility Service	–		–
	Zhong et al. [170]	The system-wide assistance service that can solve the accessibility problem of people with hand tremors	Touch area enhancement in Accessibility Service	8	Motor-impaired	Acquisition time via types of touch and trial interfaces
DEASD	Alnfiai et al. [3]	The BrailleTap calculator using tap gestures	Talkback's screen reader	2	Visually impaired	Average time spent for calculation the equation
	Chen et al. [28]	The voice assistant for fall detection, safety care, phone's accessibility, daily broadcasting information, view description via AI and mobile computing	Used the Accessibility Service		The volunteers with "wearing a blinder"	"Computing time of object detection in different internet environment"
	Correa et al. [31]	A configurable game for education	Different types of touch input (Tap, Double tap, swipe, tow finger swipe)	6	Visually impaired	Efficiency and satisfaction
	Samonte et al. [138]	A learning app incorporating haptic and a voice feedback for braille recognition and 3D printing for visually impaired students	Talkback's screen reader and user feedback (haptic, vibration)	Uncounted	Visually impaired	Success rate of tasks execution
	Ghidini et al. [74]	A electronic calendar by voice, keyboard, and touch command interaction, and identify what interaction is best for visually impaired	Talkback's screen reader	4	Visually impaired	Interviewed user experience when participant perform the proposed tasks
	Hann et al. [79]	A system of digital accessible information for print disabled people	Used and Customized Talkback	–		–
	Ganz et al. [73]	A low cost and a friendly indoor navigation system for visually impaired	Used Accessibility Service's functions: "explore by touch" and "hovering"	Uncounted	Not visually impaired	Correctness of the navigation instructions and the user interface by performing certain functions
	Pareddy et al. [119]	A system that captures and embeds the meaning of basic contents as images for visually impaired.	Used Accessibility Service to collect UI hierarchy	5	Visually impaired	Survey of X-Ray's learnability, comfort, usefulness, perceived speed, perceived accuracy, satisfaction with Likert scale of 1 to 7
CFASD	Dobosz et al. [62]	Finding the best method among existing user interface approaches for playing mobile games	Talkback's screen reader	11	Visually impaired	Interface reaction time, commands correctness, screen size influence when using mobile game
	Rodriques et al. [135]	A method to investigate smartphone adoption and usage by beginner users with visual impairments	Used Talkback to understand device usage and interaction observation	5	Visually impaired	Interviewed the main concerns, observations, challenges, barriers, and experiences about smartphones, device usage, and interaction observation
	Alshyban et al. [4]	Tools to evaluate Android accessibility's capabilities in terms of apps, developers, and users	Used the Accessibility Service to monitor the AccessibilityEvent when each app is crawled	66	Practitioners	Conduct survey of current practices, challenges, accessibility tools, and guidelines related to accessibility

et al. [170] and Rodriques et al. [133] aimed to improve the accessibility for users with disabilities who have hand tremors or suffer from speech impairment. Table 6 summarizes studies in CFASD: what the research purpose is, what functions of the AS API were used, and how usability tests were performed.

Further, according to the classification results of the research purposes in Sections 4.1 and 4.2, we identified which papers belong to each research theme and sub-theme by IP type in Table 7. To summarize, this section classified research purposes with five themes and 21 sub-themes by the thematic analysis method. Through this classification, researchers can identify overall studies that belong to each research theme and sub-theme of existing research over the past 10 years.

5 CATEGORIZATION OF ANDROID MOBILE USAGE AND SENSOR DATA

We investigate the mobile usage and sensor data that Android APIs can collect. Moreover, we present the type of mobile usage and sensor data used for each of the research purposes of the surveyed papers classified in Section 4.1.

5.1 Excluded Studies in Data Categorization

The data in IP1 studies (i.e., used AS or US API *other than* accessibility enhancement purposes) is divided into three cases as follows: (1) mobile usage and sensor data collected through user studies, (2) mobile usage and sensor data collected through other methods such as developer test, crawling, crowdsourcing other than user study, and (3) types of mobile usage and sensor data that can be

Table 7. Results of Research Purpose Categorization Via IP Types

Theme	Sub Theme	Ref: IP1-1-AS	Ref: IP1-1-US	Ref: IP1-2-AS	Ref: IP1-2-US	Ref: IP2-1	Ref: IP2-2
UP	GUPA	[19, 30, 70, 84, 94]	[125–127, 144, 159, 165]				
	MLMS	[92]	[63, 83, 163]	[164]			
	FSD	[6, 168]		[18, 72, 81, 111, 140, 141]			
NT	IFN	[25, 123, 154]	[108]				
	NT/NM	[93, 116, 117, 120, 121, 124]	[90, 91]				
	HNAU	[5, 24, 61, 85, 89, 122]					
EUI/UX	CE	[147]		[64, 148, 156]			
	UIE	[26, 97, 166]		[130, 137]	[137]		
	UIP	[104, 132]		[131]			
	IFASD					[33, 133, 134, 151, 166, 167, 170]	[29, 169]
	DEASD					[3, 28, 31, 73, 74, 119, 138]	[79]
	CFASD					[4, 62, 135]	
PS	ASS	[13, 22, 87]	[150, 171, 172]				
	APC	[20, 128]	[20]				
	PPS	[67, 88, 118]	[7]	[69]			
	MDPS	[113]		[95, 101, 129, 142, 153]	[157, 158]		
PTS	GAT	[14]		[77, 112, 139]			
	CT			[102, 161]			
	APS	[98, 99]		[96, 100]			
	PMM	[17]	[162]				
	TRSGT	[68]		[105, 114]			

collected in a framework developed for data collection purposes. We included all studies in the data categorization, except for one study [139], for which it was difficult to determine the scope of the data used. Next, among the surveyed studies corresponding to IP2 (i.e., used AS or US API for accessibility enhancement purposes), only eight surveyed studies [4, 28, 119, 133–135, 167, 170] that performed user studies and collected data are included in the data categorization. Therefore, 96 studies of the total number of surveyed studies (n = 109) are included in data categorization.

5.2 Method of Data Item Categorization

As shown in Figure 3, we create a hierarchical structure of the mobile usage and sensor data item terms using data categorization to identify and classify data collected from the third-party Android apps developed through Android APIs (e.g., AS/US APIs) in the surveyed studies. The hierarchical structure of the mobile usage and sensor data terms are created in four phases: (1) find and extract the data term list from papers, (2) create a prototype of data categorization, (3) rearrange the data categorization, and (4) reflect on issues in the categories.

5.2.1 *Find and Extract Data Term.*

A data term list of mobile usage and sensor data that can be collected through the Android framework/application developed in the prior studies is obtained. Data terms described in studies were individual words or words included in categories and sentences. Examples of each description type for the screen on/off data are as follows:

- Word form: “Screen (on, off)” [5], “Screen on/off” [25], “Screen on/off event” [30], “Screen state” [85].
- Sentence form: “Time when the phone’s screen was turned on or off” [123], “Determine whether the user is currently using the phone” [120].

5.2.2 *Devising Data Typology.*

We create a typology of data by conducting data term integration and making the hierarchical structure for the prototype of data categorization. The data term lists are grouped based on similarities. Unified data terms are defined as the representative data term for each grouped data by referring to the data term of the existing research (Table 8) and Android developer documentation [37]. In addition, a hierarchical prototype of data categorization with

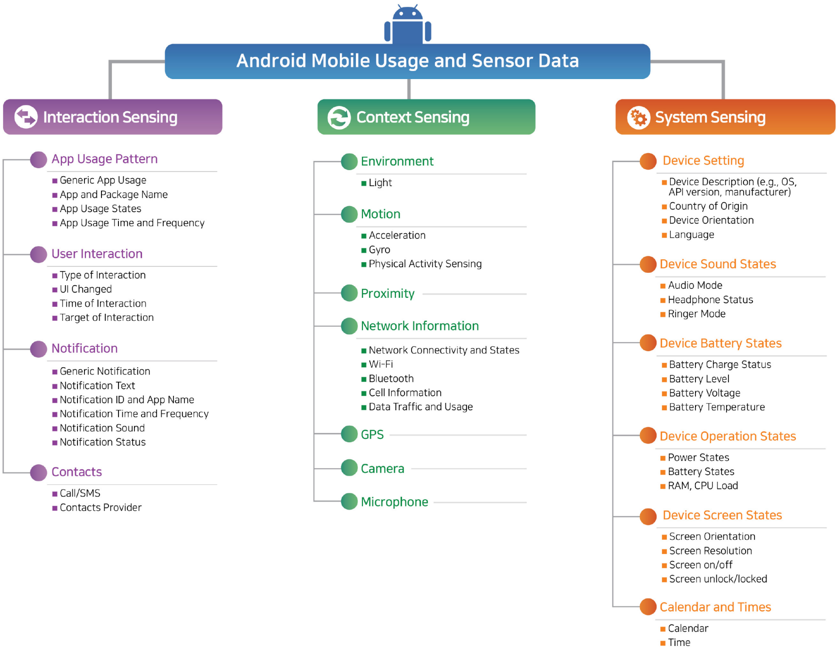

Fig. 3. Android mobile usage and sensor data categories.

five layers was created by referring to the data categorization of the existing research (Table 8) and the Android developer documentation [37].

5.2.3 Revising Data Typology. We revised the hierarchical structure when carefully reviewing the overall data categorization. The steps are as follows: (1) The data are grouped into categories or the existing categories are divided into multiple categories, reflecting on issues such as difficulties in understanding the collected mobile usage and sensor data term and in showing all data types on the chart because of too many data types; (2) the categories of data terms not described in the papers are excluded; (3) if the papers state that data are collected or can be collected, but no categories of data in the prototype of data categorization are available, then a new category is created and added in the prototype of data categorization.

Among the surveyed studies, there was a paper that stated "sensors" without specifying what sensor data is for the data type collected. We cover this issue in more detail by classifying the surveyed studies based on the degree of easy understanding of the data topology in Section 7.2.

Figure 3 shows the data categorization-based four-layer hierarchical structure results according to the unification of the mobile usage and sensor data terms described in the studies. The meaning of the second layer is as follows:

- ***Interaction sensing***: Mobile usage and sensor data that users generate by interacting with or initiating an app or system service (excluding system-specific data).
- ***Context sensing***: Mobile usage and sensor data to understand the system and user context.
- ***System sensing***: Mobile usage and sensor data on the main state of the system (e.g., resources, screen status, device settings).

To summarize, this section presented the standardized terms and classification with a four-layer hierarchical structure, which helps researchers to better understand the mobile usage and sensor

Table 8. Mobile Usage and Sensor Data Categories of Surveyed Studies

Ref.	Category	Sub-Category
Ferreira et al. [70]	"Application session"	"when, for how long and which applications were active, provided by the Accessibility Service API screen usage"
	"Screen usage"	"current screen status (on/off) and for how long it was on/off"
Ferreira et al. [72], Church et al. [30]	"Hardware sensors"	"Accelerometer, magnetometer, and photometer"
	"Software sensors"	"User's calendar, email, social activity, and other logs (e.g., calls, messages)"
	"Human-based sensors"	"mobile questionnaires (i.e., for ESM), voice, or gesture input"
Visuri et al. [154]	"Contextual information"	"location, physical activity, headphone jack, ringer mode, screen state, battery information, network information, Foreground application"
	"Notification information"	"source application contents notification outcome"
chang et al. [24]	"Notification"	"Sender application, arrival time, tag, ticker text, sound mode, vibrate mode, category, title, text, subtext, priority, visibility, action, audio mode, clearable, ongoing of a notification-flag"
	"Context"	"Lastest used, application and time, ringer mode, audio mode, volume, network and wifi status, location, physical activity, acceleration, rotation, gravity, orientation, proximity, ambient light level, battery level, charging state"
	"Accessibility"	"Triggered time, event type, event text, screen status"
Lee et al. [94]	"Application events"	"Active/inactive apps, touch and text input events, web browsing URLs, and notification events"
	"System events"	"Power on/off and screen on/off/unlock"
	"Phone events"	"Call and SMS"
Pielot et al. [121]	"Phone state"	"Ringer mode, battery level"
	"Usage"	"Screen use, app launches, and access to the notification center"
Zhang et al. [166]	"Event listeners"	"Interface accessibility events (e.g., button click, text field focus, view update, app screen switch, device app switch)"
	"Content introspections"	"Element (e.g., content, size, state, possible actions)"
	"Automation"	"Accessibility service representation (e.g., click, long press, select, scroll, or text input directed to a node in the tree)"
Schweizer et al. [141], Schweizer et al. [140]	"Physical sensor"	"Sound pressure, ring mode, acceleration, activity, illuminance, network connectivity"
	"Software sensor"	"Browsing history, contents, calendar, location, call log, foreground app"
Holzmann et al. [81]	"App usage data"	"Visited pages ("Screens" within an app), dwell times (per page), number of interaction (per page), device orientation"
	"Interaction data"	"Type of interaction (touch, scroll, long touch), interaction target"
	"Network data"	"Network type, network subtype, roaming status"
	"Device data"	"Device description, country of origin, language, network operator, operating system, screen resolution"
	"Battery data"	"Battery level, charging status (charging/discharging), temperature, voltage"
	"Context data"	"Light condition"
	"Orientation data"	"Device orientation"
Dingler et al. [61]	"Phone context"	"Ringer mode (Mute, vibration, ringer), charging mode (unplugged, charging), battery status, display orientation (portrait or landscape mode, orientation changes), light sensor, proximity sensor, location, motion"
	"Phone usage"	"Calls (incoming, outgoing), SMS (incoming, outgoing), notifications (received, dismissed, ignored, interacted with), screen (on/off events), unlocks (phone unlocks), data usage (upload/download), applications (applications in foreground, switches, usage duration)"
Anderson et al. [5]	"Notification"	"Application (name, package), Notification time (issued, interacted, posted), Notification ongoing (true, false)"
	"Contacts"	"Contact (hash), Relation (family, friend, work, none)"
	"Events"	"WiFi (SSID, BSSID), Location (latitude, longitude), Application (name, package), Time (date, issued, interacted, duration), ESM-Role (private, work), ESM-Interruptibility (private, work, both, not interruptible), Physical Activity (Google's Activity Recognition API), Power (connected, disconnected), Screen (on, off), Ringer mode (silent, vibrate, normal)"
Chang et al. [25]	"Contexual information"	"Location, activity recognition (Google activity recognition service API), sensors, network, calendar, phone status (ringer mode, screen on/off), currently running application"
Chang et al. [24]	"Notification"	"sender application, arrival time, tag, ticker text, sound mode, vibrate mode, category, title, text, subtext, priority, visibility, action, audio mode, clearable, ongoing of a notification-flag"
	"Context"	"latest used, application and time, ringer mode, audio mode, volume, network and wifi status, location, physical activity, acceleration, rotation, gravity, orientation, proximity, ambient light level, battery level, charging state"
	"Accessibility"	"triggered time, event type, event text, screen status"
Lee et al. [89]	"Contextual information"	"notifications, user's location and physical activity, activity type, the phone's status (e.g., ringer mode, battery level), sensor data, user actions"

data used in previous studies. Note that the data collected through the AS API (i.e., the main Android API in this study) can be further classified. For example, among the user interaction data corresponding to the fourth layer, the type of interaction data can be classified as click, long click, scroll, and typing once more. For the visible uniformity of the hierarchical structure, however, the three-layer hierarchical structure was mainly used in Section 6. and user interactions were further explored in Section 6.4.

6 CATEGORIZATION RESULTS OF COLLECTED DATA AND RESEARCH PURPOSES

This section illustrates data categorization results to help researchers to better understand the trend of data categories collected for various research purposes. In Section 6.1, overall results of data categorization are presented in Table 9, which includes information on which themes, sub-themes, and IP types as depicted in Table 4 for each of 96 surveyed studies. In Section 6.2, data categories used for each theme are presented in Table 9 and the analysis results are explained via Figure 4. In Section 6.3, data categories used for each research theme and sub-theme are presented in Table 10. Furthermore, data types used in each theme and sub-theme are analyzed. This process provides the results and interpretations of how the types of data collected are different according to the themes and sub-themes. Further, user interaction data are categorized and analyzed in Section 6.4.

6.1 Collected Data Categorization

This study analyzed the trends and differences in data categories used and collected by each research theme in Table 9. The columns in Table 9 are organized based on the second layer in Figure 3. Only user interaction data are based on the third layer, because they are collected by using the main API (AS API) in this review. Among the UP and NT studies, 58.3% (14 of 24) and 78% (14 of 18) collected more than one data category in the system, context, and interaction sensing data category groups, respectively; 26.3% (5 of 19), 19.0% (4 of 21), and 21.4% (3 of 14) surveyed studies belong to EUI/UX, PS, and PTS themes collected and utilize system and context sensing data categories along with interaction sensing data categories. Through this, the researchers can infer that more varied types of mobile usage and sensor data are collected in UP and NT papers than EUI/UX, PS, and PTS studies.

6.2 Research Purpose Specific Data Categorization

The number of studies that collected and used mobile usage and sensor data was identified for each research purpose as depicted in Figure 4. The frequently used data types were app usage patterns (49 studies), type of interaction (45 studies), the target of interaction (41 studies), motion (24 studies), notifications. (24 studies), device screen stats (21 studies), GPS (20 studies), UI changed (17 studies), and network information (17 studies).

Figure 4 illustrates how data are used differently depending on the themes. In UP and NP themes, all data types were used in at least more than one surveyed study. In the UP theme, app usage pattern data was the most used of all data types, with 20 studies, and more than doubled the second most used data type, the interaction data type. Because of the 22 studies using the US API, nine of them were in the UP theme, and five of them only collected app usage pattern data. In addition, the device screen states, GPS, and network information data were widely used after app usage pattern data and type of interaction data with eight, eight, and seven, respectively. Accordingly, the researchers can infer that context and system sensing data were also widely used with interaction sensing data in the UP theme. In the NT's 18 surveyed studies, all data types except device setting data were used in more than one study. The most used data were notification data, used in 18 studies. Besides, motion, device screen status, app usage patterns, device sound states, GPS, types of interaction, network information data were frequently used.

In contrast, most of the studies in EUI/UX, PS, and PTS themes mainly use interaction sensing data (i.e., user interaction, and app usage pattern). Among the 19 EUI/UX studies, target/type of interaction data was used in about 62% of studies. App usage pattern and target of interaction data were used in approximately 50% of 21 PS studies. In the PTS theme, target/type of interaction data was used in 12 (85.7%) and 9 (64.29%) of 14 studies.

6.3 Sub-theme Level Categorization

This section reviewed how many surveyed studies were used in each data category for each sub-theme and analyzed the results with three measures as follows: the number of types of data categories, the sum of utilized data categories (i.e., calculated by counting each data category used in the papers), and the average number of data categories used per surveyed study (i.e., the sum of utilized data categories/total number of surveyed studies) by each sub-theme as depicted in Table 10. To easily understand the number of data categories used by each sub-theme in Table 10, cells were marked with a darker green color as the number of papers increased. Moreover, Table 10 is displayed with dark red cells as the number increases to easily distinguish the number of papers for each sub-theme and the trend calculated by three measures. Accordingly, this study synthesized the results of Sections 6.1 and 6.2, and interpreted the trends in utilization of data categorization for each research theme/sub-theme.

Table 9. Results of Mobile Usage and Sensor Data Categorization

Ref.	Theme	Sub-Theme	Type of Included Papers	Interaction Sensing							System Sensing						Context Sensing						
					User Interaction																		
				App usage patterns	Type of interaction	UI changed	Time of interaction	Target of interaction	Notifications	Contacts information	Calendar & Standard Time	Device setting	Device sound states	Device Battery states	Device Operation states	Device screen states	Microphone	Light	Motion	Proximity	Network information	GPS	Camera
Blanke et al. [19]	UP	GUPA	IP1-1-AS	•					•												•		
Church et al. [30]	UP	GUPA	IP1-1-AS	•												•					•	•	
Kim et al. [84]	UP	GUPA	IP1-1-AS	•	•				•				•			•			•		•	•	
Lee et al. [94]	UP	GUPA	IP1-1-AS	•	•				•	•					•	•							
Ferreira et al. [70]	UP	GUPA	IP1-1-AS	•												•							
Yuan et al. [165]	UP	GUPA	IP1-1-US	•												•							
Qin et al. [125]	UP	GUPA	IP1-1-US	•																			
Qin et al. [126]	UP	GUPA	IP1-1-US	•																			
Radesky et al. [127]	UP	GUPA	IP1-1-US	•																			
Singh et al. [144]	UP	GUPA	IP1-1-US	•																			
Welke et al. [159]	UP	GUPA	IP1-1-US	•						•						•						•	
Lee et al. [92]	UP	MLMS	IP1-1-AS	•	•	•	•	•															
Khan et al. [83]	UP	MLMS	IP1-1-US	•															•			•	
Dutta et al. [63]	UP	MLMS	IP1-1-US	•																			
Yan et al. [163]	UP	MLMS	IP1-1-US	•																			
Yu et al. [164]	UP	MLMS	IP1-2-AS					•															
Andone et al. [6]	UP	FSD	IP1-1-AS	•					•	•					•	•					•	•	
Zheng et al. [168]	UP	FSD	IP1-1-AS					•															
Ferreira et al. [72]	UP	FSD	IP1-2-AS	•	•					•	•								•				•
Bitsch et al. [18]	UP	FSD	IP1-2-AS		•														•			•	
Holzmann et al. [81]	UP	FSD	IP1-2-AS	•	•		•	•				•		•	•	•		•			•		
Montague et al. [111]	UP	FSD	IP1-2-AS		•																		
Schweizer et al. [141]	UP	FSD	IP1-2-AS	•	•			•		•	•		•					•	•		•	•	
Schweizer et al. [140]	UP	FSD	IP1-2-AS	•	•			•		•	•		•					•	•		•	•	
Pielot et al. [123]	NT	IFN	IP1-1-AS						•	•			•			•				•			
Visuri et al. [154]	NT	IFN	IP1-1-AS	•					•				•	•		•			•		•	•	
Chang et al. [25]	NT	IFN	IP1-1-AS	•					•	•	•		•			•			•		•	•	
Mehrotra et al. [108]	NT	IFN	IP1-1-US	•	•				•							•			•			•	
Lee et al. [93]	NT	IFN	IP1-1-US	•	•				•							•			•			•	
Okoshi et al. [117]	NT	NT/NM	IP1-1-AS		•	•		•	•														
Okoshi et al. [116]	NT	NT/NM	IP1-1-AS		•	•		•	•														
Park et al. [120]	NT	NT/NM	IP1-1-AS		•			•	•								•		•		•		
Pielot et al. [121]	NT	NT/NM	IP1-1-AS	•					•	•			•	•		•		•	•			•	•
Pradhan et al. [124]	NT	NT/NM	IP1-1-AS	•					•		•		•	•		•	•		•		•		
Lee et al. [90]	NT	NT/NM	IP1-1-US	•					•		•								•	•			
Lee et al. [91]	NT	NT/NM	IP1-1-US	•	•	•							•			•	•						
Anderson et al. [5]	NT	HNAU	IP1-1-AS	•					•	•	•		•		•	•			•		•	•	
Chang et al. [24]	NT	HNAU	IP1-1-AS	•	•	•	•	•	•				•	•	•			•	•	•	•	•	
Dingler et al. [61]	NT	HNAU	IP1-1-AS	•					•	•			•	•		•		•	•	•	•	•	
Komuro et al. [85]	NT	HNAU	IP1-1-AS						•							•			•				
Lee et al. [89]	NT	HNAU	IP1-1-AS		•				•				•	•					•			•	
Pielot et al. [122]	NT	HNAU	IP1-1-AS						•														
Sun et al. [147]	EUI/UX	CE	IP1-1-AS		•												•						
Elazhary et al. [64]	EUI/UX	CE	IP1-2-AS		•																		
Tarakji et al. [148]	EUI/UX	CE	IP1-2-AS		•												•						
Wang et al. [156]	EUI/UX	CE	IP1-2-AS					•															
Chen et al. [26]	EUI/UX	UIE	IP1-1-AS		•			•															
Li et al. [97]	EUI/UX	UIE	IP1-1-AS	•				•															
Rauen et al. [130]	EUI/UX	UIE	IP1-2-AS	•	•									•								•	
Ryu et al. [137]	EUI/UX	UIE	IP1-2-AS/US					•															
Lin et al. [104]	EUI/UX	UIP	IP1-1-AS		•	•					•			•		•		•	•		•	•	
Riegler et al. [131]	EUI/UX	UIP	IP1-2-AS		•		•	•															
Riegler et al. [132]	EUI/UX	UIP	IP1-1-AS		•	•	•																
Zhang et al. [167]	EUI/UX	IFASD	IP2-1					•															
Zhong et al. [170]	EUI/UX	IFASD	IP2-1		•		•	•															
Rodrigues et al. [133]	EUI/UX	IFASD	IP2-1		•		•	•															
Rodrigues et al. [134]	EUI/UX	IFASD	IP2-1					•															
Chen et al. [28]	EUI/UX	DEASD	IP2-1																•		•	•	
Pareddy et al. [119]	EUI/UX	DEASD	IP2-1					•															
Rodrigues et al. [135]	EUI/UX	CFASD	IP2-1	•			•	•															
Alshayban et al. [4]	EUI/UX	CFASD	IP2-1	•	•	•		•															

(Continued)

Table 9. Continued

Ref.	Theme	Sub-Theme	Type of Included Papers	Interaction Sensing						System Sensing								Context Sensing					
					User Interaction																		
				App usage patterns	Type of interaction	UI changed	Time of interaction	Target of interaction	Notifications	Contacts information	Calendar & Standard Time	Device setting	Device sound states	Device Battery states	Device Operation states	Device screen states	Microphone	Light	Motion	Proximity	Network information	GPS	Camera
Aras et al. [13]	PS	ASS	IP1-1-AS	●																		●	
Kraus et al. [87]	PS	ASS	IP1-1-AS	●																			
Canfora et al. [22]	PS	ASS	IP1-1-AS		●														●		●		
Zhu et al. [171]	PS	ASS	IP1-1-US	●											●			●					
Zhu et al. [172]	PS	ASS	IP1-1-US	●											●			●					
Torres at al. [150]	PS	ASS	IP1-1-US	●																			
BonnÃČÂľ et al [20]	PS	APC	IP1-1-AS/US	●				●															
Rahman et al. [128]	PS	APC	IP1-1-AS		●	●		●															
Fawaz et al. [67]	PS	PPS	IP1-1-AS				●		●														
Lau et al. [88]	PS	PPS	IP1-1-AS	●	●	●																	
Ozcan et al. [118]	PS	PPS	IP1-1-AS					●															
Andriotis et al. [7]	PS	PPS	IP1-1-US	●																			
Fernandes et al. [69]	PS	PPS	IP1-2-AS		●	●		●															
Naseri et al. [113]	PS	MDPS	IP1-1-AS		●			●															
Rastogi et al. [129]	PS	MDPS	IP1-2-AS					●															
Shao et al. [142]	PS	MDPS	IP1-2-AS					●															
Li et al. [101]	PS	MDPS	IP1-2-AS	●	●	●		●															
Vishwamitra et al. [153]	PS	MDPS	IP1-2-AS					●															
Leguesse et al. [95]	PS	MDPS	IP1-2-AS					●															
Wang et al. [157]	PS	MDPS	IP1-2-US	●																			
Wang et al. [158]	PS	MDPS	IP1-2-US	●																			
Arruda et al. [14]	PTS	GAT	IP1-1-AS		●																		
Gu et al. [77]	PTS	GAT	IP1-2-AS		●			●															
Muangsiri et al. [112]	PTS	GAT	IP1-2-AS		●	●		●															
Lian et al. [102]	PTS	CT	IP1-2-AS		●	●																	
Wu et al. [161]	PTS	CT	IP1-2-AS		●			●															
Li et al. [98]	PTS	APS	IP1-1-AS	●	●			●	●														
Li et al. [99]	PTS	APS	IP1-1-AS			●		●															
Li et al. [100]	PTS	APS	IP1-2-AS		●		●	●															
Li et al. [96]	PTS	APS	IP1-2-AS		●	●		●									●						
Bissig et al. [17]	PTS	PMM	IP1-1-AS	●	●	●									●								
Xiang et al. [162]	PTS	PMM	IP1-1-US	●											●							●	
Fazzini et al. [68]	PTS	TRSGT	IP1-1-AS	●	●	●																	
Liu et al. [105]	PTS	TRSGT	IP1-2-AS		●			●															
Negara et al. [114]	PTS	TRSGT	IP1-2-AS		●			●															

6.3.1 *Usage Pattern Research Theme.* The number of data categories used of **Framework and System Development (FSD)** sub-theme studies were higher compared with other sub-themes studies in the UP theme as described in Table 10. Most studies of FSD develop third-party apps to collect various types of mobile usage and sensor data. By contrast, the studies of **Generic Usage Pattern Analysis (GUPA)** and **Machine Learning-based Model and System (MLMS)** sub-themes mainly used interaction sensing data. Accordingly, the average used data categories was 2.6, which is lower compared to FSD (6.0) as shown in Table 10. When the AS API was used among surveyed studies of GUPA and MLMS sub-themes, various mobile usage and sensor data were collected by using other Android APIs as well. However, using the US API, the average of data categories used per surveyed study was relatively low. It only needs app usage patterns about the current usage status of the device system or third-party apps rather than detailed contextual information.

6.3.2 *Notification Research Theme.* Notification theme studies collected and utilized various data categories compared to other themes. As shown in Table 10, the NT studies had higher values in three measures compared to other themes (i.e., the sum of utilized data categories, number of types of data categories, and average number of data categories used per surveyed study). In

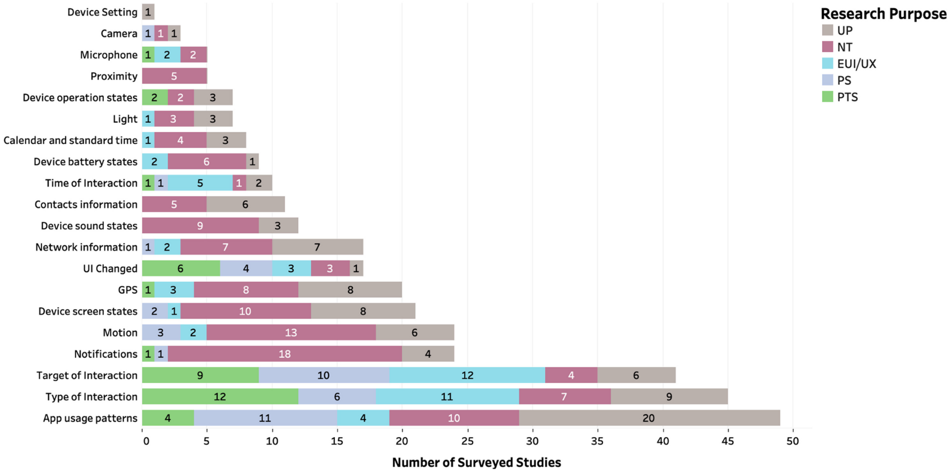

Fig. 4. Breakdown of research purposes for each mobile usage and sensor data item.

addition, the proportion of using context sensing and system sensing data along with interaction sensing data was the highest compared to other themes according to the analysis in Section 6.3. The reasons why NT theme studies have utilized various data categories are as follows. Studies of Identifying Factors respond to Notifications sub-theme utilized a variety of sensor data to predict in which context a specific notification interacts with the user [25, 123, 154]. Studies of How Notification Affects a User sub-theme collected user interaction data to understand the effects of notifications according to various situations on users [120, 121, 124]. In addition, studies of **Notification Timing and Notification Management (NT/NM)** sub-theme collected various types of mobile data to find the right notification timing and manage notification in various contexts [5, 24, 61, 85, 89].

6.3.3 EUI/UX Research Themes. Studies of EUI/UX theme collect and utilize on average 2.72 data types per a study as shown in Table 10. Since most studies in the EUI/UX theme were conducted to improve the user interface, only user interaction data were collected and analyzed by using the AS API. Only five studies collected context sensing data types along with interaction sensing data types [28, 104, 130, 147, 148].

6.3.4 Privacy and Security Research Themes. As shown in Table 10, the average number of data categories used per the surveyed study of PS theme was 2.05, which means most studies of PS theme used only a few interaction data types. The studies of **Access and Permission Control (APC)**, **Privacy-Preserving System (PPS)**, and **Monitoring and Detection for Privacy and Security (MDPS)**, which are sub-themes of PS, mainly focused on user interaction data collection and utilization. Among them, a total of 15 studies related to privacy and security collected and utilized user interaction data types such as user interaction type/target or UI changed data types [7, 20, 67, 69, 88, 95, 101, 113, 118, 128, 129, 142, 153, 157, 158]. In contrast, studies of the Authentication System/Scheme sub-theme used the context and system sensing data types along with user interaction data types to understand users' daily habits and current device hold situations or develop the motion sensor-based authentication system [13, 22, 87, 150, 171, 172].

6.3.5 Programming and Testing Support Research Themes. Most of the studies related to the PTS theme collected and utilized only a few user interaction data categories (the average

Table 10. Number of Studies That Used Based on the Types of Data in Each Research and Sub-research Theme

		Interaction sensing							**System sensing**						**Context sensing**										
		User Interaction																							
Theme	**Sub-Theme**	**App usage patterns**	**Type of Interaction**	**UI changed**	**Time of Interaction**	**Target of Interaction**	**Notifications**	**Contacts information**	**Calendar and Standard Time**	**Device setting**	**Device sound states**	**Device Battery states**	**Device Operation states**	**Device screen states**	**Microphone**	**Light**	**Motion**	**Proximity**	**Network information**	**GPS**	**Camera**	**# of types of data categories**	**Sum of utilized data categories**	**# of surveyed studies**	**Average of data categories used per surveyed study**
UP	GUPA	11	2				3	2			1		1	6			1		3	3		10	33	11	3.0
UP	MLMS	4	1	1	1	2											1			1		7	11	5	2.2
UP	FSD	5	6		1	4	1	4	3	1	2	1	2	2		3	4		4	4	1	17	48	8	6.0
NT	IFN	3	1				4	2	1		3	1		4			3	1	2	3		12	28	4	7.0
NT	NT/NM	4	4	2		3	8	1	2		2	2		3	2	1	5	2	2	1	1	17	45	8	5.6
NT	HNAU	3	2	1	1	1	6	2	1		4	3	2	3		2	5	2	3	4		17	45	6	7.5
EUI/UX	CE		3			1									2							3	6	4	1.5
EUI/UX	UIE	2	2			3						1								1		5	9	4	2.3
EUI/UX	UIP		3	2	2	1			1			1		1		1	1		1	1		11	15	3	5.0
EUI/UX	IFASD		2		2	4																3	8	4	2.0
EUI/UX	DEASD					1											1		1	1		4	4	2	2.0
EUI/UX	CFASD	2	1	1	1	2																5	7	2	3.5
PS	ASS	5	1											2			3		1		1	6	13	6	2.2
PS	APC	1	1	1		2																4	5	2	2.5
PS	PPS	2	2	2	1	2	1															6	10	5	2.0
PS	MDPS	3	2	1		6																4	12	8	1.5
PTS	GAT		3	1		2																3	6	3	2.0
PTS	CT		2	1		1																3	4	2	2.0
PTS	APS	1	3	2	1	4	1								1							7	13	4	3.3
PTS	PMM	2	1	1									2							1		5	7	2	3.5
PTS	TRSGT	1	3	1		2																4	7	3	2.3
Sum of UP		20	9	1	2	6	4	6	3	1	3	1	3	8	0	3	6	0	7	8	1	34	92	24	3.8
Sum of NT		10	7	3	1	4	18	5	4	0	9	6	2	10	2	3	13	5	7	8	1	46	118	18	6.6
Sum of EUI/UX		4	11	3	5	12	0	0	1	0	0	2	0	1	2	1	2	0	2	3	0	31	49	19	2.6
Sum of PS		11	6	4	1	10	1	0	0	0	0	0	0	2	0	0	3	0	1	0	1	20	40	21	1.9
Sum of PTS		4	12	6	1	9	1	0	0	0	0	0	2	0	1	0	0	0	0	1	0	22	37	14	2.6
Total		49	45	17	10	41	24	11	8	1	12	9	7	21	5	7	24	5	17	20	3	153	336	96	3.5

number of data categories per study was 2.62). In addition to these data, two studies have collected device operating state data for system performance measurement and management [17, 96, 162]. Except for one study using microphone data [96], studies of **GUI Automated Testing (GAT)**, **Crowdsourced Testing (CT)**, **Automated Programming Support (APS)**, and **Test Recorder and Script Generator Technique (TRSGT)** sub-themes mainly used AS API-based interaction sensing data types for crowdsourced testing, GUI automated testing, user interaction recording, automated programming support [14, 68, 77, 96, 98–100, 102, 105, 112, 114, 161].

6.4 User Interaction Level Categorization

The type of touch interaction data (e.g., click, long click, typing, scroll) were further classified and analyzed for each surveyed study. Among a total of 44 such studies, 18 studies used terms related to the types of touch interaction actions (i.e., click, long click, typing, scroll, pinch, swipe), and the 15 studies used the event name of *AccessibilityEvent* such as VIEW_CLICKED,

Table 11. Collected Touch Interaction Type Information from the Surveyed Papers

Types of Touch Interaction	Ref.	Num
General Expression	[4, 24, 72, 89, 98, 120, 128, 130, 133, 147]	10
Click (including tapping)	[14, 17, 64, 68, 69, 77, 81, 84, 92, 96, 100, 102, 104, 105, 108, 111, 112, 114, 116, 117, 131, 132, 148, 161, 170]	25
Long click	[81, 84, 92, 96, 100, 102, 108, 112, 114, 116, 117, 132, 161]	13
Typing	[18, 22, 26, 64, 68, 94, 96, 100–102, 104, 112–114, 116, 117, 140, 141, 148, 161]	20
Scroll (Including Pinch and Swipe)	[77, 81, 84, 88, 101, 102, 104, 108, 111, 112, 114, 116, 117, 131, 132, 148, 161, 170]	18

VIEW_LONG_CLICKED, and VIEW_TEXT_CHANGED, VIEW_SCOLLED to describe the collection of user's touch interaction data. The remaining 11 studies used general terms (e.g., user action, touch event, all UI events, touch event, input action, gesture action) in their papers. As shown in Table 11, these studies were grouped and classified into five categories: general expression, click (including tapping), long click, typing, and scroll (including punch and swipe). Click and typing were most frequently used in previous research, because they are the most basic interactions in smartphone use.

Among the studies using AS API, 17 studies described *AccessibilityEvent* terms in their papers. As shown in Table 12, *AccessibilityEvent* was divided into four types: view types (e.g., VIEW_CLICKED), transition types (e.g., WINDOW_STATE_CHANGED), notification types (e.g., NOTIFICATION_STATE_CHANGED), and exploration types (e.g., TOUCH_INTERACTION_START). The sum of accessibility events belonging to the four types *AccessibilityEvent* described in these 17 studies are 53, 21, 2, and 2, respectively. Accordingly, the view and transition types are described more than the notification and exploration types in the 18 studies. Although notification data were used in most NP theme studies, only two papers described notification types of *AccessibilityEvent* (NOTIFICATION_STATE_CHANGED). This tendency shows that notification types of AccessibilityEvent were not properly described in the previous papers. In addition, the exploration type was described less frequently because of its low utility rather than described in a different way in the paper. Among the *AccessibilityEvent*, the most frequent view type events were VIEW_CLICKED. In addition to 18 studies describing *AccessibilityEvent* used in Table 12, we deduced which *AccessibilityService* was used for 24 surveyed studies that used the AS API-based interaction sensing data but did not describe which types of *AccessibilityEvent* were used to detect the interaction sensing data. According to *AccessibilityEvent* that are inferred to have been used to capture the utilized data, the list of surveyed studies is as follows.

- Notification types (i.e., NOTIFICATION_STATE_CHANGED): Eighteen studies used notification data [5, 6, 19, 25, 61, 67, 85, 89, 93, 94, 98, 116, 117, 120–123, 154].
- Exploration types (i.e., TOUCH_GESTURE_DETECTION_START/END): Two studies used gesture interaction data [4, 133].
- View types (i.e., VIEW_CLICKED): Forty-two studies in Table 11 (except References [4, 108, 133]).
- Transition types (i.e., WINDOW_STATE_CHANGED): Eighteen studies used UI changed data [4, 17, 24, 68, 69, 88, 92, 96, 99, 101, 102, 104, 112, 116, 117, 128, 132] and 30 studies that used app usage pattern data [4–6, 13, 17, 19, 24, 25, 30, 61, 68, 70, 72, 81, 84, 87, 88, 92, 94, 97, 98, 101, 121, 124, 130, 135, 140, 141, 154].

7 DISCUSSION

Several issues and implications found while investigating existing studies are discussed in this section. In Section 7.1, a summary of categorization results is presented. In Section 7.2, the difficult cases to identify the smartphone data terms are summarized. In Section 7.3, the impact of changes

Table 12. Accessibility Events Described in Surveyed Studies

				VIEW TYPES								TRANSITION TYPES			NOTIFICATION TYPES	EXPLORATION TYPES	
Ref.	Theme	Sub-Theme	Type of Included Papers	CLICKED	LONG CLICKED	SELECTED	FOCUSED	TEXT CHANGED	TEXT TRAVERSED AT MOVEMENT GRANULARITY	TEXT SELECTION CHANGED	SCROLLED	WINDOW STATE CHANGED	WINDOW CONTENT CHANGED	WINDOW CHANGED	NOTIFICATION STATE CHANGED	TYPE TOUCH INTERACTION START	TYPE TOUCH INTERACTION END
Lee et al. [92]	UP	MLMS	IP1-1-AS	•	•							•					
Okoshi et al. [117]	NT	NT/NM	IP1-1-AS	•	•	•	•	•	•	•	•	•	•		•		
Okoshi et al. [116]	NT	NT/NM	IP1-1-AS	•	•	•	•	•	•	•	•	•	•		•		
Riegler et al. [132]	EUI/UX	UIP	IP1-1-AS	•	•	•	•				•	•	•	•		•	•
Rodrigues et al. [134]	EUI/UX	IFASD	IP2-1			•	•										
Aras et al. [13]	PS	ASS	IP1-1-AS										•				
Lau et al. [88]	PS	PPS	IP1-1-AS								•	•	•				
Fernandes et al. [69]	PS	PPS	IP1-2-AS	•		•							•				
Naseri et al. [113]	PS	MDPS	IP1-1-AS					•	•								
Li et al. [101]	PS	MDPS	IP1-2-AS								•		•				
Muangsiri et al. [112]	PTS	GAT	IP1-2-AS	•	•			•			•	•					
Lian et al. [102]	PTS	CT	IP1-2-AS	•	•	•			•		•			•			
Li et al. [100]	PTS	APS	IP1-2-AS	•	•			•									
Li et al. [96]	PTS	APS	IP1-2-AS	•	•	•	•	•				•	•				
Bissig et al. [17]	PTS	PMM	IP1-1-AS	•								•	•				
Fazzini et al. [68]	PTS	TRSGT	IP1-1-AS	•				•				•	•				
Liu et al. [105]	PTS	TRSGT	IP1-2-AS	•			•										

via the API version on research is reported, and the differences due to changes in API policy are discussed. In Section 7.4, privacy issues that may arise in terms of data collection in mobile usage and sensor data-driven analytics research are analyzed. In Section 7.5, factors and solutions to data quality degradation occurring in mobile usage and sensor data-driven research are briefly reviewed. In Section 7.6, limitations and future research through this review are discussed.

7.1 Summary of Categorization Results

This categorization results in this work helped researchers to understand what data was used for what research purpose. First, this research identified which data categories were used for each surveyed study via the research theme, sub-theme, and type of IP, as shown in Table 9. However, it was difficult to grasp at a glance what data was used for each research purpose through Table 9. Accordingly, this work analyzed which data categories were used for each research theme, as shown in Figure 4. The results showed that the trends of the used data types are different according to each research theme. Respectively, 83% and 58% of the NT and UP studies utilized context and system sensing data as well as interaction sensing data. Unlike NT and UP studies, prior EUI/UX, PS, and PTS studies mainly used interaction sensing data (74%, 81%, and 80%, respectively), and, thus, context and system sensing data usage was relatively low.

Further, the data types used by sub-themes classified in each research theme were analyzed to understand the data types used for each research purpose in more detail, as shown in Table 10. As a result, different trends were shown for the data types used according to the sub-themes for each research theme. In NT and UP studies, data types corresponding to context sensing or system sensing data in all sub-themes except one sub-theme (i.e., MLMS) were utilized along with data types belonging to interaction sensing data. In the EUI/UX studies, there has been at least one research using data types belonging to context sensing or system sensing data in addition to interaction sensing data in four sub-themes (i.e., **computational enhancement (CE)**, UIE, UIP, and DEASD) of six sub-themes. In the PS and PTS studies, six sub-themes (i.e., APC, PPS, MDPS, GAT, CT, and TRSGT) of nine sub-themes only used interaction sensing data. Our analysis results helped researchers and practitioners identify the tendency of data types utilized according to various research purposes and derive new insights for selecting the range of data items to be collected for research planning.

7.2 Reproducibility Risks Due to a Lack of Standardized Data Typology

While investigating the mobile usage and sensor data collected and used in the studies, the identification of mobile usage and sensor data terms in the studies can be divided into (1) the detailed explanation of data terms and (2) the vague description of data terms. As shown in Table 8, for studies of (1) in which it is easy to identify the used events and data terms are clear, and the data are categorized with the description of each data and displayed in a table. In contrast, for studies of (2) in which the API and data names are not explicitly described, or the scope of the data is unclear, because only a few examples are described. For example, it is difficult to identify what data are collected by specific papers that used vague terms (e.g., "UI Interaction Collection," "sensors," and "sensor data") rather than specifying what UI interaction data or sensor data are utilized using the AS API. Next, there are some papers where it is difficult to understand the entire data used as these papers only described a few examples using "such as," "e.g." and "ex." The specification of the collected data described in case (2) papers do not interfere with the understanding of the papers. However, if the data specification is described more specifically as in case (1), then the researcher's understanding of the data can be much clearer. In that case, it can be communicated more clearly to the readers to identify the data that are described in each paper. The studies were classified into two types according to the degree of the detailed description in the papers in terms of data categorization based on the hierarchical structure:

- Detailed described studies as the data term level of the third layer in Figure 4 (e.g., "click," "scroll," "text changed," "notification time," "accelerometer," "gyro," and "battery level") [5, 6, 14, 17, 19, 30, 61, 68, 70, 81, 84, 85, 88, 92–94, 104, 113, 114, 116, 117, 122–124, 132, 140, 141, 154].
- Vaguely described studies as the data term level of the first or second layer in Figure 4 (e.g., "all UI events," "sensors," "sensor data," "event type," "user actions," "gesture action," "context events," "context logs") [24, 25, 89, 120, 133, 139, 156, 168].

Furthermore, depending on the method of describing the data terms, there can be a difference in the degree of ease in terms of reproducibility for other researchers. The way of representing data terms in 110 papers can be divided into two types of studies: The first type of studies are described by words or words with categorization, and words in sentences describe the second type of studies. As shown in Table 8, the data term descriptions in the first type of study are more straightforward to understand than those of the second type of study. Besides, there are surveyed studies that described the Android-related component terms (e.g., API, class, event, smartphone data) as described in the Android or Google developer documentation. In addition, there are surveyed studies that did not refer to the terms in the Android or Google developer documentation. Researchers who are new to data-driven research, instead of experienced researchers who have conducted smartphone-related research, would find it easier to grasp and reproduce papers with data terms written according to Android or Google developer documentation. Therefore, if the terms used to describe events and data match with that in the Google developer documentation, then future researchers can understand the related research more quickly in this field.

7.3 Influence of API Updates on Research

This subsection reports the influence of Android APIs (e.g., AS/US API, NotificationListenerService, and NotificationManager) release and update on the research trend. As shown in Figure 5, this study derived the number of reviewed papers for each research theme via API level, year, and Android version. The releases and updates of AS/US API from 2008 to 2022 are described in detail in Appendix D Table A5. During 2008–2012, NotificationManager (API level 1) and AS API (API level 4) were published and updated (API levels 9–17) [34, 35, 56]. Major events of AS

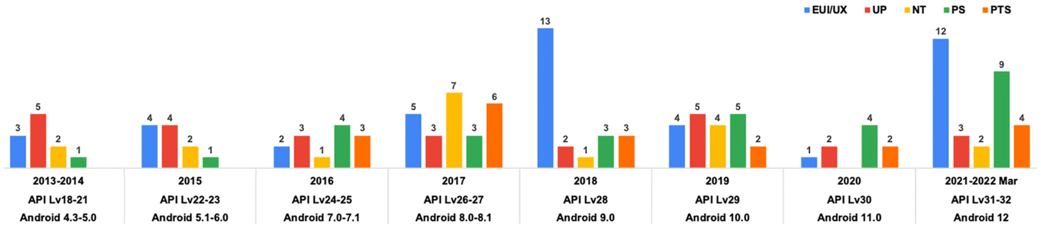

Fig. 5. Impact of Research according to API Changes History.

API were mostly updated that are view types (e.g., view clicked/long clicked/selected/focused/text changed/scroll/text selection changed), transition types (e.g., window state/content changed), exploration types (e.g., view hover enter/exit, touch interaction, touch exploration gesture), and type notification state changed at the API levels 4–17 [34].

2013–2014: US API and NotificationListenerService API Release and Start Publishing Research using AS API: According to the AS API function update in API level 1–17 [34, 35, 55, 56], the results of the study, which started in late 2011–early 2012, were first published on July 31, 2013, under LG Sangam Library's digital accessible information system for the disabled [79]. In 2014, a total of seven studies of the theme (UP: 5, EUI/UX: 2, NT: 2, PS: 1) were published in earnest based on the functions updated through API 14–17 from 2011 to 2013 as shown in Figure 5. In API 14–17, Through touch exploration mode, which provides feedback (e.g., voice, vibration) on touch content by detecting *AccessibilityEvent* [34] generated from user's gesture/touch interaction (e.g., gesture detection start/end, touch navigation, view hover enter), studies for the visually impaired corresponding to DEASD and IFASD sub-theme were published [73, 169]. Further, studies using AS API to monitor notification states changed and app usage status were published for the first time in NP theme [122, 123]. Moreover, the UP theme has three studies of GUPA sub-theme that use AS API to understand the time, type, and frequency of app usage such as smartphone overuse [19, 70, 94], and two studies under the FSD sub-theme were published that developed a framework for collecting app usage logs and ambient contextual data using smartphone sensors using SensorManager (released in API level 3) and AS API [140, 141]. Finally, the purpose of AS API-based privacy-preserving system using events such as WINDOW CONTENT CHANGED, WINDOW STATE CHANGED, and VIEW SCROLLED was published for the first time in 2014 [88]. Further, NotificationListenerService API and US API (e.g., UsageStatsManager, UsageStats, UsageEvent, UsageEvents.Event) were released in API 18 and 21 [10–12, 55].

2015–2017: Notification APIs Updates and Start Publishing Research on Various Themes: From 2015 to 2017, there was no functional update to the AS API, and only continuous updates of NotificationListenerService and NotificationManager occurred as depicted in Figure 5. There were nine studies (GAT: 3, CT: 1, APS: 2, and TRSGT: 3) corresponding to the PTS theme, and a new theme was published using AS API [14, 68, 96, 98, 105, 112, 139, 161]. Furthermore, NotificationListenerService API released in API level 18 in 2013 was first described as being utilized in a published paper corresponding to the purpose of NT/NM sub-theme of NP theme in 2017 [124]; this work used the AS API and NotificationListenerService together. NotificationListenerService was used to track more abundant notification information (e.g., post, clear, action, title, id, style, modality), and information such as notification shading or duration was used AS API.

2018–2020: Research using US APIs on the Rise: In 2018–2020, the window change-related events of AS API and the US API events (e.g., standby bucket changed/active/frequency, keyguard hidden/shown, screen interactive) were updated in API 28. In 2018, 13 studies corresponding to EUI/UX theme were published. This increment may be because the window changed related events of AS API was updated in Android 9 (released on March 7, 2018). There were also four papers on CE that aim to improve multi-modal interaction technologies such as Google Assistant (released in 2016). In addition, studies using US APIs were on the rise as well. However, these updated events of US API were not specifically used in the papers in 2019–2020. As for those papers using the US API released in API 21 in 2014, two papers (MLMS: 1, GUPA: 1) were first published in 2016 [63, 159], and US API was used in a total of five papers (NT: 1, EUI/UX: 1, UP: 2, PS: 1) in 2017–2018 [20, 108, 125, 137, 144]. The number of studies using US API was eight papers (UP: 3, NT: 1, EUI/UX: 1, PS: 3) in 2019 [91, 126, 138, 150, 157, 163, 165, 171] and six papers (UP: 2, PS: 3, PTS: 1) in 2020 [7, 83, 127, 158, 162, 172], a significant increase compared to 2015–2018. In 2018–2019 there were major updates of UsageEvent.Event (e.g., activity resumed/paused/stopped, device shut down, device start-up, and foreground service start/stop) [11] that can track app foreground, background status, and device/system status. In 2020, there were only three studies (EUI/UX: 1, PS: 1, PTS: 1) using AS API [4, 95, 99], which was far less compared to studies using US API. From 2016 to 2020, of a total of 20 studies using US API, 9 were in the UP theme, accounting for 45 % of the total. Therefore, the US API has been used in various sub-themes as a representative UP theme from 2015 to 2020. The current trend is increasingly using the US API by replacing the AS API for app usage log collection.

Latest Trends as of January 2021–March 2022: Note that our preliminary search of recent papers published in 2021/01–2022/03 (via ACM Digital Library, Scopus, Web of Science, and ScienceDirect) revealed that research themes of EUI/UX and Privacy and Security are on the rise (12 EUI/UX, 9 PS, 4 PTS, 3 UP, and 2 NT studies) as depicted in Appendix D. In API level 32 of 2021, new events (e.g., SPEECH_STATE_LISTENING/SPEAKING/CHANGED_START/END) associated with microphone's Listening and Speaking status were added [34] and expected new studies through these events.

Challenges of US/AS APIs: App usage patterns can be grasped through the US API from 2015 onwards, making it more possible to understand various status information about app usage intuitively. Nevertheless, we cannot collect the information regarding interaction types (e.g., click, long click, scroll, focused, and typing) and interaction targets (UI elements and hierarchy) from the US API. In particular, studies corresponding to specific research purposes (PTS, PS) that require view information (interaction type, interaction target) inside the app had no choice but to rely on the AS API, which has an important function. While AS API can broaden the functionality of applications, it can potentially generate security risks. Once granted the user's permissions, the API can be used to read data from other apps. Therefore, Google currently regulates the AS API use. It can be used only to improve the accessibility for users with disabilities for apps registered in the Google Play Store.[6] Suppose the developer does not follow the policy. In that case, the policy support team of the Google Play Console[7] will inform the developers to explain how the AS API is used in their developed application to help users with disabilities. If the third-party app does not meet the requirements within 30 days, then the ability to AS API within the app will be removed, or the app is regulated and unpublished [32]. So far, for apps that use AS API other than those intended to assist the users with disabilities, there are no restrictions on publishing apps to other app stores or web hosting services (e.g., GitHub). Besides, when the AS API is used for the mobile usage and sensor data-driven research, there are no specific prohibition regulations by Google. However, since the AS API has powerful functions that can fetch almost all user interaction data (e.g., interaction type/target and app status) about the app being used, the Google Android policy regulation may become more severe due to personal privacy and security issues. Hence, even

[6] https://play.google.com/store.

[7] https://support.google.com/googleplay/android-developer/answer/7218994?hl=ko.

when using AS API for future research purposes, other alternatives are needed to prepare future restrictions.

7.4 Privacy Issues in Personal Data Research

In the surveyed studies of this article, mobile usage and sensor data collected and utilized through Android AS/US API and other APIs may cause privacy issues in the process of collection, storage, and utilization. For example, those studies that analyze smartphone data in the wild to understand users' smartphone usage patterns and their surrounding contexts (i.e., interaction, context, and system sensing data) can be used to reconstruct a user's everyday life patterns. Especially, AS API can be used to hijack sensitive personal data such as credit information [60]. Furthermore, among the surveyed studies of this article, studies that collect large-scale data over a long period may further increase the risk of privacy and security. Among the UP and NT research themes, if the goal is to identify the predictive features or build machine learning models, then it would be beneficial to collect various types of smartphone data, possibly with many participants for a longer duration to improve the external validity of the research.

When conducting mobile usage and sensor data-driven research covered in this article, minimizing privacy risks is required, and this should be carefully addressed in the user consent such as **Institutional Review Board (IRB)** documents and the app's request for consent. Indeed, surveyed studies in which data were collected from multiple participants over a long period of time in the wild at risk for privacy were IRB approved, and users' informed consent was obtained prior to the experiment [72, 84, 159, 165]. However, many studies do not mention whether the IRB-approved or users' informed consent. In addition, personal data were anonymized by removing personal details, only collecting universal unique ID, hashing sensitive information to minimize privacy and security risks [6, 92]. Additionally, the collected data were stored in a local device rather than uploaded to the cloud service [18]. Furthermore, researchers can follow well-known data protection guidelines principles that must be followed when using personal and sensitive user data. Appendix B offers the guidelines that researchers should follow when collecting and utilizing data to solve these privacy and data quality issues by referring to existing data protection guidelines (e.g., Google Play Console's Policy Center, U.S. Federal Trade Commission's fair information practices principles, and EU General Data Protection Regulation).

7.5 Data Quality Issues in Mobile Usage and Sensor Data-driven Research

Another challenge in studies of mobile usage and sensor data-driven analytics is data quality problems (e.g., missing value, outlier). Among the existing reviewed studies, we further investigated the quality issues that occurred during mobile usage and sensor data collection in the UP theme [72, 83, 84, 94, 126, 127, 144]. Major quality issues were categorized into human factors and system factors, and several guidelines for quality assurance are discussed. In a long-term wild study, the duration of a participant's data fails to meet the minimum period, and missing values occur possibly due to participants' personal situations or characteristics. Missing values occurred when participants did not follow the experimental guidelines; e.g., turning off wifi or GPS, turning off the device, and changing the smartphones during the study period [84, 94, 144].

Data quality issues also arise due to system factors. Radesky et al. [127] removed 13% of the collected data not properly collected due to server and app usage logger's problems (e.g., server maintenance, not installed necessary program for data collection) [127]. In addition, Khan et al. [83] mentioned several data quality issues (1) When tracking events that occur in smartphones through the app usage logger, data collection was impossible, because Android built-in APIs were not supported in certain smartphone models. (2) Data accuracy (e.g., noise, outlier) deteriorated due to the poor quality of the sensor. (3) When the app usage logger app was installed, the battery drained quickly and the smartphone turned off easily, causing many missing values [83]. Several papers mentioned strategies on data quality assurance. For example, researchers selected the participants based on their smartphone model and Android OS version. Khan et al. [83] selected participants with the most suitable manufacturer (e.g., Samsung) in consideration of factors to improve data quality (e.g., sensor quality of smartphones, Android-APIs support for collected data, and battery persistence) [83]. Human factor issues can be addressed by carefully setting up the experimental protocols and offering participant guidelines (e.g., use of smartphones for a certain period of time, system and configuration settings to always collect data). In addition, it is recommended to periodically check whether the data are collected [84].

7.6 Limitation and Future Work

This article only surveyed the studies using APIs that collect smartphone data in Android environment. Most studies based on mobile usage and sensor data-driven analytics used Android OS excepted research led by Apple Inc. [27]. According to Nishiyama et al. [115], Android can flexibly distribute third-party apps that can access various sensors and collect mobile usage and sensor data. While iOS has limitations on accessing sensors and distributing applications. Because iOS has limitations in sensing in the background, it supports that in only a few conditions (e.g., location updates, Bluetooth operations, background fetch, audio, and remote notifications). In addition, the difference between iOS and Android lies in the data types that can be collected. On iOS, it is not easy to use APIs to track app usage patterns, touch interaction types, and UI components/hierarchies. Access to mobile sensor data such as location, Bluetooth, telephony, and activity recognition in iOS must be justified for approval; iOS does not support accessing telephony, light, and temperature data [115]. Due to these limitations, prior research mostly used Android rather than iOS. Nevertheless, the global mobile operating system share of iOS devices in 2021 was about 17.2 ± 4.2% (calculated by the mean and standard deviation of first to fourth quarter in 2021) [145]. Therefore, future survey research need to investigate mobile usage and sensor data-driven studies in the iOS environment.

Furthermore, although we have separately identified the papers that entailed user studies, this review did not investigate the experiment design such as the number of participants, experiment duration, laboratory/field testing, and questionnaire data (including experience sampling method data) for IP1 studies. In future research, a review of experiment design and procedure (e.g., participant, duration, metric, tool) will help researchers when performing user studies for different purposes. Moreover, a follow-up review needs to analyze data analysis techniques when performing mobile usage and sensor data-driven analytics. For example, recent studies on digital phenotype extensively used mobile and sensor data to automatically classify user emotions and diseases.

Mobile context data-driven research in the Android OS can be investigated using keywords different from the ones used in this article. To understand the personal daily activity logs, built-in sensors (e.g., Wi-Fi, GPS) and other APIs (e.g., Google Activity Recognition, *SensorManager*) that are frequently used to collect user context data can be included as keywords. The studies using the AS/US APIs were investigated to find studies that effectively utilized app usage patterns. However, although the app usage pattern was collected in the Android OS, studies that used their developed API names instead of names of describing related to the AS or US API were excluded from the review such as References [1, 15, 16, 23, 65, 66, 71, 143, 146, 155]. Therefore, it is expected that further, broader insights can be obtained by subsequent researchers if studies that collected the app usage patterns, despite not describing AS/US APIs in their papers, are included in the review scope.

The current work fails to capture the latest publications (January 2021–March 2022), because this research was conducted in the same year. An additional search at ACM Digital Library, Scopus, Web of Science, and ScienceDirect through the keywords in Table 3 revealed 85 papers. Applying the PRISMA criteria in Figure 2 resulted in 30 papers, mostly published at well-known venues (e.g., ACM CHI/UIST/CCS, ASSET, PACM IMWUT/HCI, and USENIX S&P). The category analysis results were discussed in Section 7.3, and the detailed statistics are reported in Appendix D.

8 CONCLUSION

This study analyzed prior studies that employed Android AS/US APIs to collect mobile usage and sensor data, which are the most representative forms of app usage data that can identify personal lifelog data. Through research purpose classification with five themes and 21 sub-themes, this analysis helps researchers to understand at a glance the existing studies conducted for the past 10 years for each specific research purpose. In addition, this study presented standardized terms and taxonomy with a four-layer hierarchical structure from 109 studies reviewed, and this taxonomy helps to improve reproducibility for researchers and to better understand what data are used in the previous studies. Furthermore, the tendency analysis of data types collected for each research purpose provides insight into reducing the cost and time for data collection and analysis. This study offers systematic guidelines for future research on Android AS/US APIs and lays a foundation for promoting research reproducibility and facilitating follow-up research on data-driven analytics with mobile usage and sensor data.

APPENDIX

As the supplementary materials, we discuss the trends in research purpose and used data via IP types in Appendix A, and the privacy and security issues in mobile usage and sensor data analytics based research in Appendix B. Appendix C describe AS/US API's events. Appendix D describes the API release and update via API level (Table A5) and categorized results of 30 papers published from January 2021 to March 2022. Further, a table of abbreviations was provided in Appendix E.

REFERENCES

[1] Saeed Abdullah, Mark Matthews, Elizabeth L. Murnane, Geri Gay, and Tanzeem Choudhury. 2014. Towards circadian computing: "Early to bed and early to rise" makes some of us unhealthy and sleep deprived. In *Proceedings of the ACM International Joint Conference on Pervasive and Ubiquitous Computing*. 673–684.

[2] Stadd Allison. 2013. 79% of people 18–44 have their smartphones with them 22 hours a day. Retrieved March 12, 2022 from http://www.adweek.com/socialtimes/smartphones/480485.

[3] Mrim Alnfiai and Srinivas Sampalli. 2017. BrailleTap: Developing a calculator based on braille using tap gestures. In *Proceedings of the International Conference on Universal Access in Human-Computer Interaction*. Springer, 213–223.

[4] Abdulaziz Alshayban, Iftekhar Ahmed, and Sam Malek. 2020. Accessibility issues in android apps: State of affairs, sentiments, and ways forward. In *Proceedings of the IEEE/ACM 42nd International Conference on Software Engineering (ICSE'20)*. IEEE, 1323–1334.

[5] Christoph Anderson, Judith S. Heinisch, Sandra Ohly, Klaus David, and Veljko Pejovic. 2019. The impact of private and work-related smartphone usage on interruptibility. In *Adjunct Proceedings of the ACM International Joint Conference on Pervasive and Ubiquitous Computing and Proceedings of the ACM International Symposium on Wearable Computers*. 1058–1063.

[6] Ionut Andone, Konrad Błaszkiewicz, Mark Eibes, Boris Trendafilov, Christian Montag, and Alexander Markowetz. 2016. Menthal: A framework for mobile data collection and analysis. In *Proceedings of the ACM International Joint Conference on Pervasive and Ubiquitous Computing: Adjunct*. 624–629.

[7] Panagiotis Andriotis and Atsuhiro Takasu. 2020. To allow, or deny? That is the question. In *Proceedings of the International Conference on Human-Computer Interaction*. Springer, 287–304.

[8] Android Developer. Retrieved March 12, 2022 from https://developers.google.com/android/reference/com/google/android/gms/location/package-summary.

[9] Android Developer. 2022. TrafficStats. Retrieved March 12, 2022 from https://developer.android.com/reference/android/net/TrafficStats.

[10] Android Developer. 2022. UsageEvents. Retrieved March 12, 2022 from https://developer.android.com/reference/android/app/usage/UsageEvents.

[11] Android Developer. 2022. UsageEvents.Event. Retrieved March 12, 2022 from https://developer.android.com/reference/android/app/usage/UsageEvents.Event.

[12] Android Developer. 2022. UsageStatsManager. Retrieved March 12, 2022 from https://developer.android.com/reference/android/app/usage/UsageStatsManager#constants_1.

[13] Shravan Aras, Chris Gniady, and Hari Venugopalan. 2019. MultiLock: Biometric-based graded authentication for mobile devices. In *Proceedings of the 16th EAI International Conference on Mobile and Ubiquitous Systems: Computing, Networking and Services*. 100–109.

[14] Filipe Arruda, Augusto Sampaio, and Flávia A. Barros. 2016. Capture & replay with text-based reuse and framework agnosticism. In *Proceedings of the International Conference on Software Engineering and Knowledge Engineering (SEKE'16)*. 420–425.

[15] Joost Asselbergs, Jeroen Ruwaard, Michal Ejdys, Niels Schrader, Marit Sijbrandij, and Heleen Riper. 2016. Mobile phone-based unobtrusive ecological momentary assessment of day-to-day mood: An explorative study. *J. Med. Internet Res.* 18, 3 (2016), e72.

[16] Nikola Banovic, Christina Brant, Jennifer Mankoff, and Anind Dey. 2014. ProactiveTasks: The short of mobile device use sessions. In *Proceedings of the 16th International Conference on Human-Computer Interaction with Mobile Devices and Services*. 243–252.

[17] Pascal Bissig, Gino Brunner, Florian Gubler, Roger Wattenhofer, and Andreas Zingg. 2018. Towards measuring real-world performance of android devices. In *Proceedings of the IEEE/ACS 15th International Conference on Computer Systems and Applications (AICCSA'18)*. IEEE, 1–8.

[18] Jó Ágila Bitsch, Roann Ramos, Tim Ix, Paula Glenda Ferrer-Cheng, and Klaus Wehrle. 2015. Psychologist in a pocket: Towards depression screening on mobile phones. In *Proceedings of the International Conference on Wearable, Micro & Nano technologies for Personalized Health (pHealth'15)*. 153–159.

[19] Tobias Blanke, Giles Greenway, Jennifer Pybus, and Mark Coté. 2014. Mining mobile youth cultures. In *Proceedings of the IEEE International Conference on Big Data (Big Data'14)*. IEEE, 14–17.

[20] Bram Bonné, Sai Teja Peddinti, Igor Bilogrevic, and Nina Taft. 2017. Exploring decision making with android's runtime permission dialogs using in-context surveys. In *Proceedings of the 13th Symposium on Usable Privacy and Security (SOUPS'17)*. 195–210.

[21] John Brooke. 1996. Sus: A "quick and dirty'usability. *Usabil. Eval. Industr.* 189 (1996).

[22] Gerardo Canfora, Paolo Di Notte, Francesco Mercaldo, and Corrado Aaron Visaggio. 2016. Silent and continuous authentication in mobile environment. In *Proceedings of the International Conference on Security and Cryptography (SECRYPT'16)*. 97–108.

[23] Larry Chan, Vedant Das Swain, Christina Kelley, Kaya de Barbaro, Gregory D. Abowd, and Lauren Wilcox. 2018. Students' experiences with ecological momentary assessment tools to report on emotional well-being. *Proc. ACM Interact. Mob. Wear. Ubiq. Technol.* 2, 1 (2018), 1–20.

[24] Yung-Ju Chang, Yi-Ju Chung, Yi-Hao Shih, Hsiu-Chi Chang, and Tzu-Hao Lin. 2017. What do smartphone users do when they sense phone notifications? In *Proceedings of the ACM International Joint Conference on Pervasive and Ubiquitous Computing and Proceedings of the ACM International Symposium on Wearable Computers*. 904–909.

[25] Yung-Ju Chang and John C. Tang. 2015. Investigating mobile users' ringer mode usage and attentiveness and responsiveness to communication. In *Proceedings of the 17th International Conference on Human-Computer Interaction with Mobile Devices and Services*. 6–15.

[26] Fanglin Chen, Kewei Xia, Karan Dhabalia, and Jason I. Hong. 2019. MessageOnTap: A suggestive interface to facilitate messaging-related tasks. In *Proceedings of the CHI Conference on Human Factors in Computing Systems*. 1–14.

[27] Richard Chen, Filip Jankovic, Nikki Marinsek, Luca Foschini, Lampros Kourtis, Alessio Signorini, Melissa Pugh, Jie Shen, Roy Yaari, Vera Maljkovic, et al. 2019. Developing measures of cognitive impairment in the real world from consumer-grade multimodal sensor streams. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. 2145–2155.

[28] Runze Chen, Zhanhong Tian, Hailun Liu, Fang Zhao, Shuai Zhang, and Haobo Liu. 2018. Construction of a voice driven life assistant system for visually impaired people. In *Proceedings of the International Conference on Artificial Intelligence and Big Data (ICAIBD'18)*. IEEE, 87–92.

[29] Sandesh Chinchole and Samir Patel. 2017. Artificial intelligence and sensors based assistive system for the visually impaired people. In *Proceedings of the International Conference on Intelligent Sustainable Systems (ICISS'17)*. IEEE, 16–19.

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

[31] Ana G. D. Correa, Laisa C. C. De Biase, Erich P. Lotto, and Roseli D. Lopes. 2018. Development and usability evaluation of an configurable educational game for the visually impaired. In *Proceedings of the IEEE Games, Entertainment, Media Conference (GEM'18)*. IEEE, 1–9.

[32] Corbin Davenport. 2017. Google Will Remove Play Store Apps That Use Accessibility Services for Anything Except Helping Disabled Users. Retrieved March 12, 2022 from https://www.androidpolice.com/2017/11/12/google-will-remove-play-store-apps-use-accessibility-services-anything-except-helping-disabled-users/.

[33] Bruna de Oliveira, Juliana Cristina Braga, and Rafael J. Pezzuto Damaceno. 2018. Application for the configuration and adaptation of the android operating system for the visually impaired. In *Proceedings of the Internet of Accessible Things*. 1–4.

[34] Android Developer. 2022. AccessibilityEvent. Retrieved March 12, 2022 from https://developer.android.com/reference/android/view/accessibility/AccessibilityEvent.

[35] Android Developer. 2022. AccessibilityService. Retrieved March 12, 2022 from https://developer.android.com/reference/android/accessibilityservice/AccessibilityService.

[36] Android Developer. 2022. ActivityManager. Retrieved March 12, 2022 from https://developer.android.com/reference/android/app/ActivityManager.

[37] Android Developer. 2022. Android Developer, Documentation for App Developer. Retrieved March 12, 2022 from https://developer.android.com/docs.

[38] Android Developer. 2022. Android.app.usage. Retrieved March 12, 2022 from https://developer.android.com/reference/android/app/usage/package-summary.

[39] Android Developer. 2022. android.bluetooth. Retrieved March 12, 2022 from https://developer.android.com/reference/android/bluetooth/package-summary.

[40] Android Developer. 2022. android.bluetooth.le. Retrieved March 12, 2022 from https://developer.android.com/reference/android/bluetooth/le/package-summary.

[41] Android Developer. 2022. android.hardware. Retrieved March 12, 2022 from https://developer.android.com/reference/android/hardware/package-summary.

[42] Android Developer. 2022. android.location. Retrieved March 12, 2022 from https://developer.android.com/reference/android/location/package-summary.

[43] Android Developer. 2022. android.view.accessibility. Retrieved March 12, 2022 from https://developer.android.com/reference/android/view/accessibility/package-summary.html.

[44] Android Developer. 2022. Bluetooth Overview. Retrieved March 12, 2022 from https://developer.android.com/guide/topics/connectivity/bluetooth.

[45] Android Developer. 2022. Broadcast Overview. Retrieved March 12, 2022 from https://developer.android.com/guide/components/broadcasts.

[46] Android Developer. 2022. Calllog. Retrieved March 12, 2022 from https://developer.android.com/reference/android/provider/CallLog.

[47] Android Developer. 2022. Context. Retrieved March 12, 2022 from https://developer.android.com/reference/android/content/Context.

[48] Android Developer. 2022. Create Your Own Accessibility Service. Retrieved March 12, 2022 from https://developer.android.com/guide/topics/ui/accessibility/service.

[49] Android Developer. 2022. Detect When Users Start or End an Activity. Retrieved March 12, 2022 from https://developer.android.com/guide/topics/location/transitions.

[50] Android Developer. 2022. Intent. Retrieved March 12, 2022 from https://developer.android.com/reference/android/content/Intent#ACTION_AIRPLANE_MODE_CHANGED.

[51] Android Developer. 2022. LocationListener. Retrieved March 12, 2022 from https://developer.android.com/reference/android/location/LocationListener.

[52] Android Developer. 2022. Locationmanager. Retrieved March 12, 2022 from https://developer.android.com/reference/android/location/LocationManager.

[53] Android Developer. 2022. Monitor the Battery Level and Charging State. Retrieved March 12, 2022 from https://developer.android.com/training/monitoring-device-state/battery-monitoring.

[54] Android Developer. 2022. NetworkStatsManager. Retrieved March 12, 2022 from https://developer.android.com/reference/android/app/usage/NetworkStatsManager.

[55] Android Developer. 2022. NotificationListenerService. Retrieved March 12, 2022 from https://developer.android.com/reference/android/location/LocationListener.

[56] Android Developer. 2022. NotificationManager. Retrieved March 12, 2022 from https://developer.android.com/reference/android/app/NotificationManager.

[57] Android Developer. 2022. Platform Architecture. Retrieved March 12, 2022 from https://developer.android.com/guide/platform.

[58] Android Developer. 2022. Sensors Overview. Retrieved March 12, 2022 from https://developer.android.com/guide/topics/sensors/sensors_overview.

[59] Android Developer. 2022. TelephonyManager. Retrieved March 12, 2022 from https://developer.android.com/reference/android/telephony/TelephonyManager.

[60] Wenrui Diao, Yue Zhang, Li Zhang, Zhou Li, Fenghao Xu, Xiaorui Pan, Xiangyu Liu, Jian Weng, Kehuan Zhang, and XiaoFeng Wang. 2019. Kindness is a risky business: On the usage of the accessibility APIs in android. In *Proceedgins of the 22nd International Symposium on Research in Attacks, Intrusions and Defenses (RAID'19)*. 261–275.

[61] Tilman Dingler, Dominik Weber, Martin Pielot, Jennifer Cooper, Chung-Cheng Chang, and Niels Henze. 2017. Language learning on-the-go: Opportune moments and design of mobile microlearning sessions. In *Proceedings of the 19th International Conference on Human-computer Interaction with Mobile Devices and Services*. 1–12.

[62] Krzysztof Dobosz and Jakub Ptak. 2016. How to control a mobile game. In *Proceedings of the International Conference on Computers Helping People with Special Needs*. Springer, 523–529.

[63] Anandi Dutta and Magdy Bayoumi. 2016. Introducing a novel smart design framework for a reconfigurable multiprocessor systems-on-chip (MPSoC) architecture. In *Proceedings of the IEEE International Conference on Smart Computing (SMARTCOMP'16)*. IEEE, 1–3.

[64] Hanan H. Elazhary and Sahar F. Sabbeh. 2018. The w 5 framework for computation offloading in the internet of things. *IEEE Access* 6 (2018), 23883–23895.

[65] Paul Eskes, Marco Spruit, Sjaak Brinkkemper, Jacob Vorstman, and Martien J. Kas. 2016. The sociability score: App-based social profiling from a healthcare perspective. *Comput. Hum. Behav.* 59 (2016), 39–48.

[66] Hossein Falaki, Ratul Mahajan, Srikanth Kandula, Dimitrios Lymberopoulos, Ramesh Govindan, and Deborah Estrin. 2010. Diversity in smartphone usage. In *Proceedings of the 8th International Conference on Mobile Systems, Applications, and Services*. 179–194.

[67] Kassem Fawaz, Kyu-Han Kim, and Kang G. Shin. 2016. Privacy vs. reward in indoor location-based services. *Proc. Priv. Enhanc. Technol.* 2016, 4 (2016), 102–122.

[68] Mattia Fazzini, Eduardo Noronha De A. Freitas, Shauvik Roy Choudhary, and Alessandro Orso. 2017. Barista: A technique for recording, encoding, and running platform independent android tests. In *Proceedings of the IEEE International Conference on Software Testing, Verification and Validation (ICST'17)*. IEEE, 149–160.

[69] Earlence Fernandes, Oriana Riva, and Suman Nath. 2016. Appstract: On-the-fly app content semantics with better privacy. In *Proceedings of the 22nd Annual International Conference on Mobile Computing and Networking*. 361–374.

[70] Denzil Ferreira, Jorge Goncalves, Vassilis Kostakos, Louise Barkhuus, and Anind K. Dey. 2014. Contextual experience sampling of mobile application micro-usage. In *Proceedings of the 16th International Conference on Human-computer Interaction with Mobile Devices and Services*. 91–100.

[71] Denzil Ferreira, Vassilis Kostakos, Alastair R. Beresford, Janne Lindqvist, and Anind K. Dey. 2015. Securacy: An empirical investigation of android applications' network usage, privacy and security. In *Proceedings of the 8th ACM Conference on Security and Privacy in Wireless and Mobile Networks*. 1–11.

[72] Denzil Ferreira, Vassilis Kostakos, and Anind K. Dey. 2015. AWARE: Mobile context instrumentation framework. *Front. ICT* 2 (2015), 6.

[73] Aura Ganz, James M. Schafer, Yang Tao, Carole Wilson, and Meg Robertson. 2014. PERCEPT-II: Smartphone based indoor navigation system for the blind. In *Proceedings of the 36th Annual International Conference of the IEEE Engineering in Medicine and Biology Society*. IEEE, 3662–3665.

[74] Eduardo Ghidini, Wagner D. L. Almeida, Isabel H. Manssour, and Milene S. Silveira. 2016. Developing apps for visually impaired people: Lessons learned from practice. In *Proceedings of the 49th Hawaii International Conference on System Sciences (HICSS'16)*. IEEE, 5691–5700.

[75] Google. 2022. Activity Recognition API. Retrieved March 12, 2022 from https://developers.google.com/location-context/activity-recognition.

[76] Google. 2022. TalkBack. Retrieved March 12, 2022 from https://support.google.com/accessibility/android/topic/3529932?hl=en&ref_topic=9078845.

[77] Tianxiao Gu, Chengnian Sun, Xiaoxing Ma, Chun Cao, Chang Xu, Yuan Yao, Qirun Zhang, Jian Lu, and Zhendong Su. 2019. Practical GUI testing of android applications via model abstraction and refinement. In *Proceedings of the IEEE/ACM 41st International Conference on Software Engineering (ICSE'19)*. IEEE, 269–280.

[78] Michael Gusenbauer. 2019. Google scholar to overshadow them all? Comparing the sizes of 12 academic search engines and bibliographic databases. *Scientometrics* 118, 1 (2019), 177–214.

[79] Hye-Yeon Hann, Yoon-Seok Jeong, and Won-Hong Jang. 2013. Best practices of the DAISY standards in Korea-LG Sangnam library's service for the print disabled.

[80] Karen Holtzblatt, Jessamyn Burns Wendell, and Shelley Wood. 2004. *Rapid Contextual Design: A How-to Guide to Key Techniques for User-centered Design*. Elsevier.

[81] Clemens Holzmann, Dustin Steiner, Andreas Riegler, and Christian Grossauer. 2017. An android toolkit for supporting field studies on mobile devices. In *Proceedings of the 16th International Conference on Mobile and Ubiquitous Multimedia*. 473–479.

[82] IDC. 2021. Smartphone Market Share. Retrieved March 12, 2022 from https://www.idc.com/promo/smartphone-market-share/os.

[83] Soban Ahmed Khan, Asma Ahmad Farhan, Labiba Gillani Fahad, and Syed Fahad Tahir. 2020. Personal productivity monitoring through smartphones. *J. Amb. Intell. Smart Environ.* (2020), 1–15.

[84] Inyeop Kim, Rihun Kim, Heepyung Kim, Duyeon Kim, Kyungsik Han, Paul H. Lee, Gloria Mark, and Uichin Lee. 2019. Understanding smartphone usage in college classrooms: A long-term measurement study. *Comput. Educ.* 141 (2019), 103611.

[85] Kyohei Komuro, Yuichiro Fujimoto, and Kinya Fujita. 2017. Relationship between worker interruptibility and work transitions android.app.usage by smartphone. In *Proceedings of the International Conference on Human-Computer Interaction.* Springer, 687–699.

[86] Lampros C. Kourtis, Oliver B. Regele, Justin M. Wright, and Graham B. Jones. 2019. Digital biomarkers for alzheimer's disease: The mobile/wearable devices opportunity. *NPJ Digit. Med.* 2, 1 (2019), 1–9.

[87] Lydia Kraus, Robert Schmidt, Marcel Walch, Florian Schaub, and Sebastian Möller. 2017. On the use of emojis in mobile authentication. In *Proceedings of the IFIP International Conference on ICT Systems Security and Privacy Protection.* Springer, 265–280.

[88] Billy Lau, Simon Chung, Chengyu Song, Yeongjin Jang, Wenke Lee, and Alexandra Boldyreva. 2014. Mimesis aegis: A mimicry privacy shield–a system's approach to data privacy on public cloud. In *Proceedings of the 23rd USENIX Security Symposium (USENIX Security'14).* 33–48.

[89] Hao-Ping Lee, Kuan-Yin Chen, Chih-Heng Lin, Chia-Yu Chen, Yu-Lin Chung, Yung-Ju Chang, and Chien-Ru Sun. 2019. Does who matter? Studying the impact of relationship characteristics on receptivity to mobile IM messages. In *Proceedings of the CHI Conference on Human Factors in Computing Systems.* 1–12.

[90] Jemin Lee, Jinse Kwon, and Hyungshin Kim. 2018. Reducing smartwatch users' distraction with convolutional neural network. *Mob. Inf. Syst.* (2018).

[91] Jemin Lee, Uichin Lee, and Hyungshin Kim. 2019. PASS: Reducing redundant interactions between a smartphone and a smartwatch for energy saving. *IEEE Trans. Mob. Comput.* (2019).

[92] Seokjun Lee, Rhan Ha, and Hojung Cha. 2018. Click sequence prediction in android mobile applications. *IEEE Trans. Hum.-Mach. Syst.* 49, 3 (2018), 278–289.

[93] Seungchul Lee, Saumay Pushp, Chulhong Min, and Junehwa Song. 2018. Exploring relationship-aware dynamic message screening for mobile messengers. In *Proceedings of the ACM International Joint Conference and 2018 International Symposium on Pervasive and Ubiquitous Computing and Wearable Computers.* 134–137.

[94] Uichin Lee, Joonwon Lee, Minsam Ko, Changhun Lee, Yuhwan Kim, Subin Yang, Koji Yatani, Gahgene Gweon, Kyong-Mee Chung, and Junehwa Song. 2014. Hooked on smartphones: An exploratory study on smartphone overuse among college students. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems.* 2327–2336.

[95] Yonas Leguesse, Mark Vella, Christian Colombo, and Julio Hernandez-Castro. 2020. Reducing the forensic footprint with android accessibility attacks. In *Proceedings of the International Workshop on Security and Trust Management.* Springer, 22–38.

[96] Toby Jia-Jun Li, Amos Azaria, and Brad A. Myers. 2017. SUGILITE: Creating multimodal smartphone automation by demonstration. In *Proceedings of the CHI Conference on Human Factors in Computing Systems.* 6038–6049.

[97] Toby Jia-Jun Li, Igor Labutov, Xiaohan Nancy Li, Xiaoyi Zhang, Wenze Shi, Wanling Ding, Tom M. Mitchell, and Brad A. Myers. 2018. APPINITE: A multi-modal interface for specifying data descriptions in programming by demonstration using natural language instructions. In *Proceedings of the IEEE Symposium on Visual Languages and Human-Centric Computing (VL/HCC'18).* IEEE, 105–114.

[98] Toby Jia-Jun Li, Yuanchun Li, Fanglin Chen, and Brad A. Myers. 2017. Programming IoT devices by demonstration using mobile apps. In *Proceedings of the International Symposium on End User Development.* Springer, 3–17.

[99] Toby Jia-Jun Li, Marissa Radensky, Justin Jia, Kirielle Singarajah, Tom M. Mitchell, and Brad A. Myers. 2020. Interactive task and concept learning from natural language instructions and GUI demonstrations. In *Proceedings of the AAAI-20 Workshop on Intelligent Process Automation (IPA'20).*

[100] Toby Jia-Jun Li and Oriana Riva. 2018. KITE: Building conversational bots from mobile apps. In *Proceedings of the 16th Annual International Conference on Mobile Systems, Applications, and Services.* 96–109.

[101] Yafei Li, Jiageng Chen, and Anthony T. S. Ho. 2018. Chatting application monitoring on android system and its detection based on the correlation test. In *Proceedings of the Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC'18).* IEEE, 1556–1563.

[102] Hao Lian, Zemin Qin, Hangcheng Song, and Tieke He. 2018. E-CAT: Evaluating crowdsourced android testing. In *Proceedings of the International Conference of Pioneering Computer Scientists, Engineers and Educators.* Springer, 493–504.

[103] Yunji Liang, Xiaolong Zheng, and Daniel D. Zeng. 2019. A survey on big data-driven digital phenotyping of mental health. *Inf. Fusion* 52 (2019), 290–307.

[104] Syuan-Yi Lin and Chung-Ta King. 2019. User-centered context-aware CPU/GPU power management for interactive applications on smartphones. In *Proceedings of the 16th ACM International Conference on Computing Frontiers*. 247–250.

[105] Kuei-Chun Liu, Yu-Yu Lai, and Ching-Hong Wu. 2017. A mechanism of reliable and standalone script generator on android. In *Proceedings of the IEEE International Conference on Software Testing, Verification and Validation Workshops (ICSTW'17)*. IEEE, 372–374.

[106] Jory MacKay. 2019. Screen Time Stats 2019: Here's How Much You Use Your Phone during the Workday. Retrieved March 12, 2022 from https://blog.rescuetime.com/screen-time-stats-2018/.

[107] Alberto Martín-Martín, Enrique Orduna-Malea, Mike Thelwall, and Emilio Delgado López-Cózar. 2018. Google scholar, web of science, and scopus: A systematic comparison of citations in 252 subject categories. *J. Informetr.* 12, 4 (2018), 1160–1177.

[108] Abhinav Mehrotra, Sandrine R. Müller, Gabriella M. Harari, Samuel D. Gosling, Cecilia Mascolo, Mirco Musolesi, and Peter J. Rentfrow. 2017. Understanding the role of places and activities on mobile phone interaction and usage patterns. *Proc. ACM Interact. Mob. Wear. Ubiq. Technol.* 1, 3 (2017), 1–22.

[109] Huasong Meng, Vrizlynn L. L. Thing, Yao Cheng, Zhongmin Dai, and Li Zhang. 2018. A survey of android exploits in the wild. *Comput. Secur.* 76 (2018), 71–91.

[110] David Moher, Alessandro Liberati, Jennifer Tetzlaff, Douglas G. Altman, Prisma Group, et al. 2009. Preferred reporting items for systematic reviews and meta-analyses: The PRISMA statement. *PLoS Med.* 6, 7 (2009), e1000097.

[111] Kyle Montague, André Rodrigues, Hugo Nicolau, and Tiago Guerreiro. 2015. TinyBlackBox: Supporting mobile in-the-wild studies. In *Proceedings of the 17th International ACM SIGACCESS Conference on Computers & Accessibility*. 379–380.

[112] Woramet Muangsiri and Shingo Takada. 2017. Random GUI testing of android application using behavioral model. *Int. J. Softw. Eng. Knowl. Eng.* 27, 09n10 (2017), 1603–1612.

[113] Mohammad Naseri, Nataniel P. Borges, Andreas Zeller, and Romain Rouvoy. 2019. AccessiLeaks: Investigating privacy leaks exposed by the android accessibility service. *Proc. Priv. Enhanc. Technol.* 2019, 2 (2019), 291–305.

[114] Stas Negara, Naeem Esfahani, and Raymond Buse. 2019. Practical android test recording with espresso test recorder. In *Proceedings of the IEEE/ACM 41st International Conference on Software Engineering: Software Engineering in Practice (ICSE-SEIP'19)*. IEEE, 193–202.

[115] Yuuki Nishiyama, Denzil Ferreira, Yusaku Eigen, Wataru Sasaki, Tadashi Okoshi, Jin Nakazawa, Anind K. Dey, and Kaoru Sezaki. 2020. iOS crowd–sensing won't hurt a bit!: AWARE framework and sustainable study guideline for iOS platform. In *Proceedings of the International Conference on Human-Computer Interaction*. Springer, 223–243.

[116] Tadashi Okoshi, Hiroki Nozaki, Jin Nakazawa, Hideyuki Tokuda, Julian Ramos, and Anind K. Dey. 2016. Towards attention-aware adaptive notification on smart phones. *Perv. Mob. Comput.* 26 (2016), 17–34.

[117] Tadashi Okoshi, Julian Ramos, Hiroki Nozaki, Jin Nakazawa, Anind K. Dey, and Hideyuki Tokuda. 2015. Attelia: Reducing user's cognitive load due to interruptive notifications on smart phones. In *Proceedings of the IEEE International Conference on Pervasive Computing and Communications (PerCom'15)*. IEEE, 96–104.

[118] Ahmet Talha Ozcan, Can Gemicioglu, Kaan Onarlioglu, Michael Weissbacher, Collin Mulliner, William Robertson, and Engin Kirda. 2015. Babelcrypt: The universal encryption layer for mobile messaging applications. In *Proceedings of the International Conference on Financial Cryptography and Data Security*. Springer, 355–369.

[119] Sujeath Pareddy, Anhong Guo, and Jeffrey P. Bigham. 2019. X-Ray: Screenshot accessibility via embedded metadata. In *Proceedings of the 21st International ACM SIGACCESS Conference on Computers and Accessibility*. 389–395.

[120] Chunjong Park, Junsung Lim, Juho Kim, Sung-Ju Lee, and Dongman Lee. 2017. Don't bother me. I'm socializing! A breakpoint-based smartphone notification system. In *Proceedings of the 2017 ACM Conference on Computer Supported Cooperative Work and Social Computing*. 541–554.

[121] Martin Pielot, Bruno Cardoso, Kleomenis Katevas, Joan Serrà, Aleksandar Matic, and Nuria Oliver. 2017. Beyond interruptibility: Predicting opportune moments to engage mobile phone users. *Proc. ACM Interact. Mob. Wear. Ubiq. Technol.* 1, 3 (2017), 1–25.

[122] Martin Pielot, Karen Church, and Rodrigo De Oliveira. 2014. An in-situ study of mobile phone notifications. In *Proceedings of the 16th International Conference on Human-Computer Interaction with Mobile Devices & Services*. 233–242.

[123] Martin Pielot, Rodrigo De Oliveira, Haewoon Kwak, and Nuria Oliver. 2014. Didn't you see my message? Predicting attentiveness to mobile instant messages. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. 3319–3328.

[124] Swadhin Pradhan, Lili Qiu, Abhinav Parate, and Kyu-Han Kim. 2017. Understanding and managing notifications. In *Proceedings of the IEEE Conference on Computer Communications (INFOCOM'17)*. IEEE, 1–9.

[125] Xiangang Qin, Chee-Wee Tan, Effie Lai-Chong Law, Mads Bødker, Torkil Clemmensen, Hequn Qu, and Diandi Chen. 2018. 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 6th International Symposium of Chinese CHI*. 39–48.

[126] Xiangang Qin, Dechuan Wang, Torkil Clemmensen, Diandi Chen, and Yanuan Zhao. 2019. The association of actual behavior on wechat with psychological traits: Evidence with mixed method. In *Proceedings of the 7th International Symposium of Chinese CHI*. 49–56.

[127] Jenny S. Radesky, Heidi M. Weeks, Rosa Ball, Alexandria Schaller, Samantha Yeo, Joke Durnez, Matthew Tamayo-Rios, Mollie Epstein, Heather Kirkorian, Sarah Coyne, et al. 2020. Young children's use of smartphones and tablets. *Pediatrics* (2020).

[128] Md Lutfor Rahman, Ajaya Neupane, and Chengyu Song. 2018. Iac: On the feasibility of utilizing neural signals for access control. In *Proceedings of the 34th Annual Computer Security Applications Conference*. 641–652.

[129] Vaibhav Rastogi, Rui Shao, Yan Chen, Xiang Pan, Shihong Zou, and Ryan Riley. 2016. Are these ads safe: Detecting hidden attacks through the mobile app-web interfaces. In *Proceedings of the Network and Distributed System Security (NDSS'16)*.

[130] Zachary Rauen, Fazel Anjomshoa, and Burak Kantarci. 2018. Empowering human-computer interaction in securing smartphone sensing. In *Proceedings of the IEEE 23rd International Workshop on Computer Aided Modeling and Design of Communication Links and Networks (CAMAD'18)*. IEEE, 1–6.

[131] Andreas Riegler and Clemens Holzmann. 2015. UI-CAT: Calculating user interface complexity metrics for mobile applications. In *Proceedings of the 14th International Conference on Mobile and Ubiquitous Multimedia*. 390–394.

[132] Andreas Riegler and Clemens Holzmann. 2018. Measuring visual user interface complexity of mobile applications with metrics. *Interact. Comput.* 30, 3 (2018), 207–223.

[133] André Rodrigues. 2015. Breaking barriers with assistive macros. In *Proceedings of the 17th International ACM SIGACCESS Conference on Computers & Accessibility*. 351–352.

[134] André Rodrigues, Leonardo Camacho, Hugo Nicolau, Kyle Montague, and Tiago Guerreiro. 2018. Aidme: Interactive non-visual smartphone tutorials. In *Proceedings of the 20th International Conference on Human-Computer Interaction with Mobile Devices and Services Adjunct*. 205–212.

[135] André Rodrigues, Kyle Montague, Hugo Nicolau, and Tiago Guerreiro. 2015. Getting smartphones to talkback: Understanding the smartphone adoption process of blind users. In *Proceedings of the 17th International ACM SIGACCESS Conference on Computers & Accessibility*. 23–32.

[136] John Rooksby, Alistair Morrison, and Dave Murray-Rust. 2019. Student perspectives on digital phenotyping.

[137] Woo-Jong Ryu, HyeonTaek Oh, and SangKeun Lee. 2017. sigInterface: A personalized interface for intelligent services. In *Proceedings of the ACM International Joint Conference on Pervasive and Ubiquitous Computing and Proceedings of the ACM International Symposium on Wearable Computers*. 185–188.

[138] M. J. Samonte, E. D. Laurente, K. M. Magno, and C. Perez. 2019. Braille3D: Using haptic and voice feedback for braille recognition and 3D printing for the blind. *J. Manage. Sci. Eng.* 482, 1 (2019), 012027.

[139] Jose Lorenzo San Miguel and Shingo Takada. 2016. Gui and usage model-based test case generation for android applications with change analysis. In *Proceedings of the 1st International Workshop on Mobile Development*. 43–44.

[140] Immanuel Schweizer, Roman Bärtl, Benedikt Schmidt, Fabian Kaup, and Max Mühlhäuser. 2014. Kraken. me mobile: The energy footprint of mobile tracking. In *Proceedings of the 6th International Conference on Mobile Computing, Applications and Services*. IEEE, 82–89.

[141] Immanuel Schweizer and Benedikt Schmidt. 2014. Kraken. me: Multi-device user tracking suite. In *Proceedings of the ACM International Joint Conference on Pervasive and Ubiquitous Computing: Adjunct Publication*. 853–862.

[142] Rui Shao, Vaibhav Rastogi, Yan Chen, Xiang Pan, Guanyu Guo, Shihong Zou, and Ryan Riley. 2018. Understanding in-app ads and detecting hidden attacks through the mobile app-web interface. *IEEE Trans. Mob. Comput.* 17, 11 (2018), 2675–2688.

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

[144] Aman Kr Singh, Ashish Kr Prajapati, Vikash Kumar, and Subhankar Mishra. 2017. Usage analysis of mobile devices. *Proc. Comput. Sci.* 122 (2017), 657–662.

[145] Statista. 2022. Market Share Held by Apple iOS Operating System of Smartphone Shipments from 1st Quarter 2011 to 4th Quarter 2021. Retrieved March 12, 2022 from https://www.statista.com/statistics/236031/market-share-of-ios-in-global-smartphone-os-shipments/.

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

[147] Ke Sun, Chun Yu, Weinan Shi, Lan Liu, and Yuanchun Shi. 2018. Lip-interact: Improving mobile device interaction with silent speech commands. In *Proceedings of the 31st Annual ACM Symposium on User Interface Software and Technology*. 581–593.

[148] Ahmad Bisher Tarakji, Jian Xu, Juan A. Colmenares, and Iqbal Mohomed. 2018. Voice enabling mobile applications with UIVoice. In *Proceedings of the 1st International Workshop on Edge Systems, Analytics and Networking.* 49–54.

[149] Markus Tober. 2011. PubMed, sciencedirect, scopus or google scholar–which is the best search engine for an effective literature research in laser medicine? *Med. Laser Appl.* 26, 3 (2011), 139–144.

[150] José Torres, Sergio de los Santos, Efthimios Alepis, and Constantinos Patsakis. 2019. Behavioral biometric authentication in android unlock patterns through machine learning. In *Proceedings of the International Conference on Information Systems Security and Privacy (ICISSP'19).* 146–154.

[151] Daniel Trindade, André Rodrigues, Tiago Guerreiro, and Hugo Nicolau. 2018. Hybrid-brailler: Combining physical and gestural interaction for mobile braille input and editing. In *Proceedings of the CHI Conference on Human Factors in Computing Systems.* 1–12.

[152] Niels Van Berkel, Denzil Ferreira, and Vassilis Kostakos. 2017. The experience sampling method on mobile devices. *ACM Comput. Surv.* 50, 6 (2017), 1–40.

[153] Nishant Vishwamitra, Xiang Zhang, Jonathan Tong, Hongxin Hu, Feng Luo, Robin Kowalski, and Joseph Mazer. 2017. MCDefender: Toward effective cyberbullying defense in mobile online social networks. In *Proceedings of the 3rd ACM on International Workshop on Security And Privacy Analytics.* 37–42.

[154] Aku Visuri, Niels van Berkel, Tadashi Okoshi, Jorge Goncalves, and Vassilis Kostakos. 2019. Understanding smartphone notifications' user interactions and content importance. *Int. J. Hum.-Comput. Stud.* 128 (2019), 72–85.

[155] Rui Wang, Fanglin Chen, Zhenyu Chen, Tianxing Li, Gabriella Harari, Stefanie Tignor, Xia Zhou, Dror Ben-Zeev, and Andrew T. Campbell. 2014. StudentLife: Assessing mental health, academic performance and behavioral trends of college students using smartphones. In *Proceedings of the ACM International Joint Conference on Pervasive and Ubiquitous Computing.* 3–14.

[156] Ru Wang, Shizhan Chen, Zhiyong Feng, and Keman Huang. 2018. A client microservices automatic collaboration framework based on fine-grained APP. In *Proceedings of the IEEE International Conference on Services Computing (SCC'18).* IEEE, 25–32.

[157] Yingjie Wang, Xing Liu, Weixuan Mao, and Wei Wang. 2019. DCDroid: Automated detection of SSL/TLS certificate verification vulnerabilities in android apps. In *Proceedings of the ACM Turing Celebration Conference-China.* 1–9.

[158] Yingjie Wang, Guangquan Xu, Xing Liu, Weixuan Mao, Chengxiang Si, Witold Pedrycz, and Wei Wang. 2020. Identifying vulnerabilities of SSL/TLS certificate verification in android apps with static and dynamic analysis. *J. Syst. Softw.* (2020), 110609.

[159] Pascal Welke, Ionut Andone, Konrad Blaszkiewicz, and Alexander Markowetz. 2016. Differentiating smartphone users by app usage. In *Proceedings of the ACM International Joint Conference on Pervasive and Ubiquitous Computing.* 519–523.

[160] Michael Winnick. 2016. Putting a Finger on Our Phone Obsession. Retrieved March 12, 2022 from https://blog.dscout.com/mobile-touches.

[161] Guoquan Wu, Yuzhong Cao, Wei Chen, Jun Wei, Hua Zhong, and Tao Huang. 2017. AppCheck: A crowdsourced testing service for android applications. In *Proceedings of the IEEE International Conference on Web Services (ICWS'17).* IEEE, 253–260.

[162] Zhengzhe Xiang, Shuiguang Deng, Javid Taheri, and Albert Zomaya. 2020. Dynamical service deployment and replacement in resource-constrained edges. *Mob. Netw. Appl.* 25, 2 (2020), 674–689.

[163] Kaige Yan, Jingweijia Tan, and Xin Fu. 2019. Bridging mobile device configuration to the user experience under budget constraint. *Perv. Mob. Comput.* 58 (2019), 101023.

[164] Che-Hsuan Yu, Hung-Yuan Chen, Fang-Yie Leu, and Yao-Chung Fan. 2019. Understanding mobile user intent using attentive sequence-to-sequence RNNs. In *Proceedings of the International Wireless Internet Conference.* Springer, 51–61.

[165] Nalingna Yuan, Heidi M. Weeks, Rosa Ball, Mark W. Newman, Yung-Ju Chang, and Jenny S. Radesky. 2019. How much do parents actually use their smartphones? Pilot study comparing self-report to passive sensing. *Pediatr. Res.* 86, 4 (2019), 416–418.

[166] Xiaoyi Zhang, Anne Spencer Ross, Anat Caspi, James Fogarty, and Jacob O. Wobbrock. 2017. Interaction proxies for runtime repair and enhancement of mobile application accessibility. In *Proceedings of the CHI Conference on Human Factors in Computing Systems.* 6024–6037.

[167] Xiaoyi Zhang, Anne Spencer Ross, and James Fogarty. 2018. Robust annotation of mobile application interfaces in methods for accessibility repair and enhancement. In *Proceedings of the 31st Annual ACM Symposium on User Interface Software and Technology.* 609–621.

[168] Ming-Yi Zheng, Hung-Yuan Chen, Huan Chen, and Yao-Chung Fan. 2017. On cleaning and organizing context logs for mobile user profiling. In *Proceedings of the 12th International Conference on Digital Information Management (ICDIM'17).* IEEE, 161–164.

[169] Yu Zhong, T. V. Raman, Casey Burkhardt, Fadi Biadsy, and Jeffrey P. Bigham. 2014. JustSpeak: Enabling universal voice control on Android. In *Proceedings of the 11th Web for All Conference*. 1–4.

[170] Yu Zhong, Astrid Weber, Casey Burkhardt, Phil Weaver, and Jeffrey P. Bigham. 2015. Enhancing android accessibility for users with hand tremor by reducing fine pointing and steady tapping. In *Proceedings of the 12th Web for All Conference*. 1–10.

[171] Tiantian Zhu, Zhengyang Qu, Haitao Xu, Jingsi Zhang, Zhengyue Shao, Yan Chen, Sandeep Prabhakar, and Jianfeng Yang. 2019. RiskCog: Unobtrusive real-time user authentication on mobile devices in the wild. *IEEE Trans. Mob. Comput.* 19, 2 (2019), 466–483.

[172] Tiantian Zhu, Zhengqiu Weng, Guolang Chen, and Lei Fu. 2020. A hybrid deep learning system for real-world mobile user authentication using motion sensors. *Sensors* 20, 14 (2020), 3876.

Received 11 May 2021; revised 30 March 2022; accepted 6 April 2022