Personal Information Classification on Aggregated Android Application’s Permissions

Md Mehedi Hassan Onik 1, Chul-Soo Kim 1, Nam-Yong Lee 2 and Jinhong Yang 3,*

1 Department of Computer Engineering, Inje University, Gimhae 50834, Korea; hassan@oasis.inje.ac.kr (M.M.H.O.); charles@inje.ac.kr (C.-S.K.)
2 Department of Applied Mathematics, Inje University, Gimhae 50834, Korea; nylee@inje.ac.kr
3 Department of Healthcare and IT, Inje University, Gimhae 50834, Korea
* Correspondence: jinhong@inje.ac.kr; Tel.: +82-10-8518-4555

Received: 21 August 2019; Accepted: 17 September 2019; Published: 24 September 2019

Abstract: Android is offering millions of apps on Google Play-store by the application publishers. However, those publishers do have a parent organization and share information with them. Through the ‘Android permission system’, a user permits an app to access sensitive personal data. Large-scale personal data integration can reveal user identity, enabling new insights and earn revenue for the organizations. Similarly, aggregation of Android app permissions by the app owning parent organizations can also cause privacy leakage by revealing the user profile. This work classifies risky personal data by proposing a threat model on the large-scale app permission aggregation by the app publishers and associated owners. A Google-play application programming interface (API) assisted web app is developed that visualizes all the permissions an app owner can collectively gather through multiple apps released via several publishers. The work empirically validates the performance of the risk model with two case studies. The top two Korean app owners, seven publishers, 108 apps and 720 sets of permissions are studied. With reasonable accuracy, the study finds the contact number, biometric ID, address, social graph, human behavior, email, location and unique ID as frequently exposed data. Finally, the work concludes that the real-time tracking of aggregated permissions can limit the odds of user profiling.

Keywords: privacy; PII; Android; data aggregation; security; machine learning; app publisher; IoT

1. Introduction

The proliferation of the personal data breach for gaining valuable insight into user preferences has become a common phenomenon in the data-driven industrial revolution. Popular services (software, app and website) are being reliant on the personal data to provide commercial services that generate lucrative revenue streams in the business. Since the media content consumption trend is concentrated on mobile devices, the risk of personal data leakage in this environment is getting higher. Once, a company receives personal data, it not only just analyzes data but also trades to other companies as a business policy [1]. Once compromised, invaders can use the data to regain personally identifiable information (PII) and potential personally identifiable information (PPII) or partial identity to harm the users by social engineering attacks. Therefore, personal data has already become a new oil and privacy threat modeling are at the crux of the next industrial revolution [2]. Besides, recent data breaching incidents also demand further research in the privacy domain [3–6].

Mobile apps (Android and iOS) also cause a threat to user privacy [7–14]. App user identities are frequently compromised even with consensus [15,16]. Usually, apps collect user consent to access sensitive personal data in lieu of app usage. However, identity profiling by app producing companies via ‘app permissions’ is nothing new [7,9,17,18], rather personal data collected via app permissions’
are frequently breaching user’s PII [14,19]. As today’s mobile devices are facilitated with advanced
sensors, apps are allowed to collect diverse personal data [20–23].

1.1. Problem Statement and Motivation

While using apps, users are less likely to consider the actual owner of the app (publisher and owner). 
In addition, users also prefer to pick apps from the same company. However, companies often exchange
personal data, which is even more within the subsidiaries of the identical parent association [24–28].
Simultaneous closure of key Facebook subsidiaries (Facebook, Instagram, Messenger, WhatsApp,
etc.) also proves the use of identical storage or infrastructure [29,30]. Therefore, it is highly likely to
share the app permissions among linked publishers and owners [31–34]. Therefore, this exchange of
sensitive Android permissions might amplify identity leakage [35–38]. Tracking and managing of
the shared data through multiples apps are quite challenging for the users. Thus, it is substantially
important to consider the data aggregations caused re-identification at the ultimate app owners’ end
(Figure 1). Therefore, a concrete classification of PII on the above issue is highly needed to reduce the
risk of specific PII leakage [23]. Existing privacy risk-models do not consider this data aggregation
fact [20,39–41]. The study believes our work is the first in the context of PII classification where
aggregation of app permission by the same owner (publisher and owner) is measured. At one side, the
users demand higher services from apps while on the other hand, personal data flow should also be
limited, so to lessen this large-scale data re-identification threat further study is needed.

![Figure 1. Application permission aggregation caused identity revealing by the parent organization.](image)

1.2. Key Consideration and Contribution

In response to the aforementioned issues, the study analyzes how permissions can be efficiently
managed to reduce PII leakage. Figure 1 provides an overall idea of the app permission aggregation
associated identity threat model. Partial information is collected through multiple apps, so the parent
organization (PO) can analyze and infer user identities by combining that partial information. A parent
organization re-identifies a user identity by combining different parts of information that has been
collected from the individual app (App) publishes through several publishers (P; Figure 1).

With 2.6 million apps and 95 billion downloads, Android is the top mobile app platform [42].
Therefore, this study considers all the permissions owned by the top two South Korean Android app
publishers such as Kakao [43] and Naver [44]. Each of them owns several publishers to publish varieties
of apps. This study considers five publishers of Kakao [43] and two of Naver [44]. In total 720 rows
of permission sets made with 26 dangerous Android permissions of 118 apps are used. Finally, 720
permission sets gathered by app owners are classified into 8 PII classes. A web app (‘Privacy Analysis
on Mobile App’ [45]) is developed by integrating the Google Play-store application programming
interface (API) to analyze the permission aggregation caused PII threat. The key resolutions of the
study are (a) to address how Android app permissions collected by an identical owner are exposing user PII, (b) to classify PII on aggregated Android permissions using the machine learning technique and (c) to offer suggestions for reducing large-scale permission re-identification.

1.3. Roadmap

The rest of this study is designed as follows. Section 2 describes the ‘Android permission system’ associated jargons and existing works. Section 3 proposes the aggregated Android permission associated personal information risk model. Section 4 validates the proposed model with the case studies. Section 5 discusses the pros and cons of the risk model along with the future scope and recommendations. Finally, Section 6 concludes the work followed by necessary references.

2. Related Work

2.1. Personal Information and Privacy

Personal data is something that reveals an individual’s identity or features. PII is a popular jargon used in the USA, Korea and Australia [46]. The European Union uses ‘personal information’ [47]. The National Institute of Standards and Technology (NIST) [48] defines PII as "any information about an individual maintained by an agency, including (PII) any information that can be used to distinguish or trace an individual’s identity, such as name, social security number, date and place of birth, mother’s maiden name, or biometric records; and (linked PII or PPII) any other information that is linked or linkable to an individual, such as medical, educational, financial, and employment information". Studies [49–51] have also mentioned that a combination of partial identities can produce a full identity. Therefore, throughout the study, partial identity and PPII were considered as a similar object. [48–51] defined partial identity or PPII as “A partial identity is a subset of attribute values of a complete identity, where a complete identity is the union of all attribute values of all identities of this person”. Table 1 shows a few examples of PII and PPII [37,46,52,53]. De-identification (Did) is a method by which information is modified in a way that data can no longer directly or indirectly indicate an identity. After the Did, the data is known as non-personal identifiable information (NPII). The data that is unable to distinguish any identity individually is NPII (shareable and risk-free information). Australian privacy law and practice [54] defines it as “non-identifiable data, which have never been labeled with individual identifiers or from which identifiers have been permanently removed, and by means of which no specific individual can be identified. A subset of nonidentifiable data is those that can be linked with other data, so it can be known that they are about the same data subject, although the person’s identity remains unknown”. Finally, re-identification (Rid) of personal data is done by matching the anonymous identities with available partial information to expose the exact identity of the data owner [55]. In the Android ecosystem, app owners (publisher and owner) Rid user data in order to generate app user’s unique identity. Our previous works [35,37,56] have already stated, Android app permissions affect PII. Beside phones, permissions are being collected by the apps run on a smart vehicle, TV, smart car, smart toys, Internet of Things (IoT) device, etc.

Table 1. Personal information (personally identifiable information (PII) and potential personally identifiable information (PPII)).

| PII | PPII |
| --- | --- |
| Full name and home address, biological identities, contact number, full email address, national identification number, passport number, social security number, taxpayer number, patient identification data, Iris data, fingerprints, credit card info, digital identity | Partial birthdate, part of a name, state or city, job status, web cookie, few digits of SSN, employment info, medical info, blood pressure data, education status, financial info, religion, supported team, postal code, music choice, IP address, place of birth, political view, race |
2.2. Application Permission System

Both Apple (iOS; Figure 2a) and Google (Android; Figure 2b) collect user consent while using sensitive user data (Figure 2) [57,58]. As this study only focused on Android, the study domain was limited to the Android app. An Android app requires user data, and to get that Android introduced the app’s ‘permission system’. Until now Android introduced four types of permission gathering system such as normal, signature, dangerous and special. Dangerous Android permission collects user consent that allows an app to access sensitive personal data of a user [59] (Table 2).

![Screenshot of an app permission gathering system: (a) Apple (iOS) and (b) Google (Android).](image)

**Figure 2.** Screenshot of an app permission gathering system: (a) Apple (iOS) and (b) Google (Android).

**Table 2.** Android permission (dangerous Android permissions), reproduced with permission from [59], Google Developers, 2018.

| Permission Group (10) | Individual Dangerous Permissions (26) |
|----------------------|--------------------------------------|
| Calendar             | Read calendar and write calendar     |
| Call log             | Read call log, write call log and process outgoing calls |
| Camera               | Camera                               |
| Contacts             | Read contacts, write contacts and get accounts |
| Location             | Access fine location and access coarse location |
| Microphone           | Record audio                         |
| Phone                | Read phone state, read phone numbers, call phone, answer phone calls, add voice mail and use session initiation protocol (sip) |
| Sensor               | Body sensors                         |
| SMS                  | Send SMS, receive SMS, receive WAP push and receive mms |
| Storage              | Read external storage and write external storage |

2.3. Android Application Permission Associated Privacy Risk

Android boosts its privacy by changing its ‘permission system’ in the last update [59]. The latest Android version needs runtime permissions from users, which are the least acceptable. However, almost 50% of the current Android system collects ‘dangerous permissions’ at the installation time [60]. Android [37] defines it as “Dangerous permissions cover areas where the app wants data or resources that involve the user’s private information, or could potentially affect the user’s stored data or the operation of other apps.” According to Android, ‘dangerous permission’ collects sensitive data accessing permission. The app user can either choose not to use the app or allow the ‘dangerous permissions’ to access personal data, so, ‘dangerous permissions’ must be handled sophisticatedly [61]. The study classifies
the risky PII only on the collective availability of those ‘dangerous permissions’ to an app owner (Table 2).

For instance, calendar stores personal data like routine, activity and behavior. Call logs and phone status can reveal a user’s family, friends and connected peoples’ contacts [10]. Similarly, camera, microphone and sensor collect very sensitive biological and voice data. Location collects not only the exact physical position but also coarse location [62]. Finally, storage and SMS are considered as an excessive source of personal data as they store group photo, photo id, audio clip, video clip, etc. [63].

As classification helps to differentiate attack surfaces and prepare accordingly [64], studies [7,9,10,17,65] have already analyzed app permissions for privacy threat modeling [66–68]. A study [69] states five ways of privacy breaching through the app permission system. They are: (a) Consistent access of the apps to the local storage, (b) allowing more permissions than needed, (c) hiding reason for data gathering, (d) collection of data without any context to the app and (e) app developer’s illiteracy.

Another study [70] analyzed 300 state-of-the-art research collected from IEEE, ACM, Springer, ScienceDirect, etc. associated with Android security and privacy. It identified PII exposure via permission as one of the key sources of privacy leaking through Android apps.

Li et al. [71] identified ‘significant Android permissions’ out of long app permissions and proposed ‘significant permissions’. It identified 22 vital permissions through a machine learning-based method to upgrade overall privacy. However, the study overlooked the permission caused a threat to PII.

Arora et al. [72] introduced both permission (static) and network (dynamic) analysis-based privacy awareness methodology. This study also used machine learning methods to classify malicious Android samples. The study mentioned that the static (app permission) risks are often ignored by researchers, so our proposed study focused on ‘dangerous Android permission’ associated risks.

Fritsch et al. [35] proposed that partial identities can be generated from app permissions. The study considered an experimental survey and successfully proved a combination of ‘dangerous Android permissions could form PII. The author classifies sensitive personal information based on the survey output. However, the study explores permission leakage only to the app level, instead, our study also considers publishers’ and owners’ scope of data aggregation.

Liang et al. [38] detect Android malware by considering permissions declared in the app manifest. The study examines the AndroidManifest.xml file and combines information of Android permissions to classify malware and benign apps. The study implements ‘droid combine’, which accumulates and analyzes six Android permission files to enhance the security of the Android system.

Shuba et al. [73] classified PII leakage on the mobile device with AntShield [39]. It performs deep packet inspection (DPI) in the middle of the network with machine learning to find PII leakage with prior knowledge. It explicitly considers the potential data aggregation among mobile devices, which was overlooked by similar other studies [74,75]. However, legal issues demotivate the DPI method [76].

He et al. [7] analyzed over 150 Apps and found that the deployment of third-party libraries in apps might put user identities in danger. It revealed third-party apps are collecting permissions without complying with Android privacy terms. With the same notion, our study considers publishers and owners (third parties) that might threat user privacy.

Although the General Data Protection Regulation (GDPR) [47], the NIST [48] and the Health Insurance Portability and Accountability Act (HIPAA) [77] are safeguarding privacy, somehow user identities are very often re-identified by the service providers.

Finally, few of our works [36,37,78] have already stated permission aggregation can easily Rid user identity. Those studies also stated the necessity of PII risk classification on Android permission aggregation to lower Rid risk. To propose an efficient PII classification, this study considers the chance of large-scale permission aggregation by the service providers (publishers and owners).
3. Risk Modeling

This model aimed to identify the risk of personal information exposing due to user profiling by the Android app. While designing the risk model, the study considered the permission aggregation caused identity exposing threat. This section first summarizes the proposed method followed by a discussion on the Android permission system (Section 3.1), personal information types (Section 3.2) and the risk model (Section 3.3) in the light of Sections 3.1 and 3.2. The study used five major entities (Figure 3).

1. Application (app) permissions: This study considered sensitive information sensing ‘Android app permissions’. An app provides services in exchange for those permissions.
2. Android permission aggregation: This study considered the accumulation of the available Android permissions by apps, associated app publishers and the ultimate app owners.
3. Permission generates PII: This study listed PII from several established resources. The study blended a heuristic approach with a real-life survey to link the app’s permissions to a specific PII.
4. Classification of PII on aggregated permissions: After coupling a specific permission set to PII, PII was classified. This study used several machine learning approaches to verify the classification.
5. Identification of risky PII: The study identified the classified PII as a risky PII set.
6. The risk model considered the odds of permission (partial information) flow from an Android app to the publishers and the ultimate owner. Many Android apps are being published by multiple subsidiaries. As most of those subsidiaries (publishers) are owned by very few companies, the chance of permission aggregation is high. This study considered the earlier stated facts to explore a set of a risky PII classes by proposing a risk model. Since gaining a specific PII requires a precise set of Android permissions, a classification of PII on Android permissions are highly needed. In short, the risk model identified easily exposed PII caused by the Android permission aggregation-based user profiling.

![Overall architecture of the proposed study.](image)

**Figure 3.** Overall architecture of the proposed study.

3.1. Android Permission Scope Model

This section defines the personal information distribution likelihood on the Android platform. Android platform associated stakeholders and their permission gathering mechanism (user consent) were discussed here. The key stakeholders of the Android ecosystem were defined here:

**Definition 1.** Application, a software intended to perform special function and tasks of individuals through a handphone, smart TV, IoT device, smart vehicle, etc. on the Android platform. The set of applications or apps are represented by \( APP = \{A_1, A_2, \ldots, A_{NAPP}\} \), where \( N_{APP} \) are the elements of \( APP \).
**Definition 2.** Publisher, a group of sub-organizations that publishes one or more Android applications on the Google Play-store intended for the Android platform. The set of publishers are represented by $\text{PUB} = \{W_1, W_2, \ldots, W_{N_{PUB}}\}$, where $N_{PUB}$ are the elements of the set PUB.

**Definition 3.** Owner, a company that owns few subsidiaries (sub-organizations) or publishers who publish applications. The set of owners are represented by $\text{OWN} = \{K_1, K_2, \ldots, K_{N_{OWN}}\}$, where $N_{OWN}$ are the elements of the set OWN.

**Definition 4.** The purpose of an Android apps’ permission is to safeguard the personal information of an Android user. An Android application demands user consent (install and run time) to access sensitive user data via an app. Permissions, a group of personal information accessing consent, frequently gathered by Android apps during an install and run time. The set of permissions are represented by $\text{PER} = \{P_1, P_2, \ldots, P_{N_{PER}}\}$, where $N_{PER}$ are the elements of the set PER.

Out of all permissions, Android marked 26 dangerous permissions that involve the user’s private information or affect the user’s identity (requires user consent to access) [59] (Table 2). The study only considers those 26 dangerous permissions that are represented by $\{P_{n1}, P_{n2}, \ldots, P_{n26}\}$.

Naturally, $\{A_1, A_2, \ldots, A_{N_{APP}}\}$ are linked with the respective $\{W_1, W_2, W_3, \ldots, W_{N_{PUB}}\}$ and associated $\{K_1, K_2, \ldots, K_{N_{OWN}}\}$. This study now describes the data sharing scope of applications, publishers and owners.

- Firstly, $\{A_1, A_2, \ldots, A_{N_{APP}}\}$ collects four types of user consents: Normal, dangerous, signature and system [79]. Only $\{P_{n1}, P_{n2}, \ldots, P_{n26}\}$ are used by the apps for sensitive personal data collection.

- Secondly, $\{W_1, W_2, \ldots, W_{N_{PUB}}\}$ publishes and owns the $\{A_1, A_2, \ldots, A_{N_{APP}}\}$ those are available at the Google Play-store [57]. As a $\{W_1, W_2, \ldots, W_{N_{PUB}}\}$ controls the $\{A_1, A_2, \ldots, A_{N_{APP}}\}$, it can easily accumulate partial data through underneath $\{P_{n1}, P_{n2}, \ldots, P_{n26}\}$ of those $\{A_1, A_2, \ldots, A_{N_{APP}}\}$.

- Finally, $\{K_1, K_2, \ldots, K_{N_{OWN}}\}$ owns several app publishing publishers or sub-organizations. Personal information selling to the same owners and even different owners are well-practiced [12,36,80]. Therefore, $\{K_1, K_2, \ldots, K_{N_{OWN}}\}$ can directly or indirectly collect personal data from the $\{W_1, W_2, \ldots, W_{N_{PUB}}\}$ and the $\{A_1, A_2, \ldots, A_{N_{APP}}\}$ through associated $\{P_{n1}, P_{n2}, \ldots, P_{n26}\}$.

Greater data possession can lead to stronger data mining. Thus, the study provides three data distribution associated assumptions by the applications, publishers and owners respectively.

**Assumption 1.** Actually, not all $\{P_{n1}, P_{n2}, \ldots, P_{n26}\}$ are concurrently needed by an application. Each Android application demands a specific set of dangerous permissions. For example, application $A_1$ might require a set of dangerous permissions, $\{P_{n1}, P_{n5}, P_{n6}, P_{n12}, P_{n13}, P_{n16}, P_{n23}, P_{n26}\}$; application $A_2$ might require a fully different set of dangerous permissions, $\{P_{n2}, P_{n3}, P_{n8}, P_{n15}, P_{n19}, P_{n21}, P_{n23}\}$ and application $A_3$ might require another set of dangerous permissions, $\{P_{n1}, P_{n3}, P_{n8}, P_{n13}, P_{n16}, P_{n21}, P_{n25}\}$. The above example indicates that more than one application may be used to collect all available $\{P_{n1}, P_{n2}, \ldots, P_{n26}\}$. Therefore, the study inferred the possibility of dangerous permissions aggregation from multiple apps.

**Assumption 2.** Android releases several apps on the Play-store through the publishers. Usually, each publisher issues multiple applications. Naturally, a publisher uses a common data repository in order to serve underneath services [30]. In addition, a publisher does not have any legal obstacles that refrain from data aggregation out of the published applications [80,81]. Therefore, the $\{W_1, W_2, \ldots, W_{N_{PUB}}\}$ automatically gains control over the dangerous permissions of those applications. The study also inferred the possibility of $\{P_{n1}, P_{n2}, \ldots, P_{n26}\}$ aggregation by an app publisher from all of their published apps.

**Assumption 3.** For brand recognition, financial consideration and capital raising giant companies often possess subsidiaries and sub-organizations. Those subsidiaries and sub-organization also can be Android application publishers. Alike assumption 2, $\{K_1, K_2, \ldots, K_{N_{OWN}}\}$ also has no legal difficulties that inspire the parent organization for data aggregation out of several android publishers and associated applications. Therefore, the
Appl. Sci. 2019, 9, 3997 8 of 24

... conjectured the likelihood of \{P_{n1}, P_{n2}, \ldots, P_{n26}\} aggregation by an app owner of an associated app publisher and from all of their published apps.

3.2. The Personal Information Scope Model

Firstly, information usability must be defined to estimate the scope of personal information. PII, also known as significant personal information (SPI), is a vast theme. This study only focused on the PII and SPI from the privacy leaking aspects. Based on the personal data composed from well-established institutes, a detail explanation of available data types are stated here:

**Definition 5.** Direct identifiers can directly identify the user’s individuality (described in Section 2.1). This study stated that as PII = \{S_1, S_2, \ldots, S_{NPII}\}, where NPII are the elements of PII.

**Definition 6.** Quasi identifier or partial identity can jointly identify user identity (described in Section 2.1). This study termed that as PPII = \{Z_1, Z_2, \ldots, Z_{NPPII}\}, N_{PII} are the elements of PPII.

**Definition 7.** Non-personal data no longer indicates a user’s identity (described in Section 2.1). This study termed that as NPII = \{N_1, N_2, \ldots, N_{NNPII}\}, N_{PII} are the elements of NPII.

**Definition 8.** De-identification or anonymization refers to the process by which PII \{S_1, S_2, \ldots, S_{NPII}\} and PPII \{Z_1, Z_2, \ldots, Z_{NPPII}\} are eliminated from the information to convert data into NPII \{N_1, N_2, \ldots, N_{NNPII}\}. This study denoted the de-identification method as Did (described in Section 2.1).

Based on Definitions 5–8, this study stated a few more assumptions on data characteristics.

**Assumption 4.** By nature, PII and PPII are not alike (PII ≠ PPII). Therefore, the study assumed that a single PPII could never create or indicate the PII.

**Assumption 5.** Alternatively, a set of capable PPIIs (linked attributes) can generate or indicate a specific PII. At least, two or more PPIIs are needed to generate or leak a specific PII.

Now, the correlation between dangerous Android permission and personal data will be discussed.

3.3. Android Permission Triggered User Profiling

This section related dangerous Android permissions and PII to create the permission to the PII generation model (permission versus PII matrices). Based on Sections 3.1 and 3.2, a few assumptions are stated.

**Assumption 6.** \{P_{n1}, P_{n2}, \ldots, P_{n26}\} carries sensitive personal data and is never de-identified (Did) during collecting, storing and sharing. Therefore, \{P_{n1}, P_{n2}, \ldots, P_{n26}\} is either PII or PPII, but surely not NPII.

**Assumption 7.** Again, \{P_{n1}, P_{n2}, \ldots, P_{n26}\} does not carry enough information to be direct identifiers or PII. Therefore, to be more precise, \{P_{n1}, P_{n2}, \ldots, P_{n26}\} are actually PPII.

From the discussion of Assumptions 1–7 and Definitions 1–8, the study had the following remarks

**Remark 1.** From the aforementioned assumptions 1–7 and definitions 1–8, the study concluded that a set of \{P_{n1}, P_{n2}, \ldots, P_{n26}\} were capable of \{S_1, S_2, \ldots, S_{NPII}\} or PII formation.

**Remark 2.** Assumption 1–3 and Remark 1 proves that the \{W_1, W_2, \ldots, W_{NPUB}\} are very much capable of \{S_1, S_2, \ldots, S_{NPII}\} formation (user identity leak) from the \{P_{n1}, P_{n2}, \ldots, P_{n26}\} collected through several \{A_1, A_2, \ldots, A_{NAPP}\} published by the multiple \{W_1, W_2, \ldots, W_{NPUB}\}.

Therefore, the first objective of the study, large scale Android permission aggregation by the publishers and owners, was proved by remark 2. Remark 2 was further illustrated here with an
example. If $\delta$ is the required information that can create a PII, Figure 4a shows that several individual PPII could provide enough information that led to $\delta$. Similarly, Figure 4b shows how owner X could generate that $\delta$ from three applications (A, B and C) of two publishers (J and K). While doing so, app A, B and C collected personal data through P₁, P₂, P₉ and P₁₀ and P₁₅, P₁₆, P₁₉, P₂₅ and P₂₆ respectively for the publisher J and K. Afterward, owner X gathered all the data required to get $\delta$ from publisher J and K. Finally, owner $X'$ owned the $\delta$ that could identify a PII of an application user. In short, $A \cup B \cup C \cup J \cup K \subset \delta$ or PII was owned by the $X$.

**Remark 3.** Only a particular group of PPIIs generates a specific PII, similarly only a specific set of dangerous permissions generate a specific PII. The second objective of the study, 'classification of PII's according to aggregated dangerous permissions' was done by creating a dangerous permission versus PII matrix. The study classified for which particular set of $\{P_{n1}, P_{n2} \ldots P_{n26}\}$, a specific $\{S_1, S_2 \ldots S_N\}$ was threatened. Table 3 demonstrates an example of the second study objective. A company $X'$ could re-identity four PII through four combinations of a dangerous permissions set (Table 3). As a different set of dangerous permissions led us to a different PII, this study separated those permission sets to classify the correlations (which PII was leaked for a certain permission set). Four PII's ($\alpha$, $\beta$, $\delta$ and $\gamma$) of a user were attained by $X'$ if permission set (5th, 10th, 15th and 23rd); (1st, 15th, 9th and 26th); (3rd, 4th, 9th and 17th) and (5th, 6th, 23rd and 26th) were owned (aggregated) by the owner $X'$ respectively. Sections 3.3.1 and 3.3.2 explained the proposed PII classification in detail.

![Figure 4.](image_url) **Figure 4.** (a) PII generation from a set of PPII and (b) PII generation from aggregated permissions.

**Table 3.** Dangerous Android permissions set generating PII.

| Case | Dangerous Android Permissions (1st to 26th) Owned by an Owner Company | Generated PII |
|------|-----------------------------------------------------------------------|---------------|
| Set 1| 5th 10th 15th 23rd                                                   | $\alpha$     |
| Set 2| 1st 15th 9th 26th                                                   | $\beta$      |
| Set 3| 3rd 4th 9th 17th                                                   | $\delta$     |
| Set 4| 5th 6th 23rd 26th                                                   | $\gamma$     |

### 3.3.1. Connection Between Android Permission and PPII

Since Android permissions can convey significant partial information directly or indirectly to a company, therefore, this study considered those app permissions as PPII. Tables 2 and 4 describe the Android permission system in detail. This study provided enough evidence to assume that the PPII (Table 1) and the dangerous Android permission (Table 2) was a single entity [35–37,56,82,83].
Table 4. Detail description of dangerous Android permissions, reproduced with permission from [59], Google Developers, 2019.

| Dangerous Android Permission | Dangerous Android Permissions       | Detail Description                                                                 |
|------------------------------|------------------------------------|------------------------------------------------------------------------------------|
| 1                            | read_calendar                      | Read and write user information from the calendar                                   |
| 2                            | write_calendar                     |                                                                                    |
| 3                            | camera                             | Access camera and capture image and video                                          |
| 4                            | read_contacts                      |                                                                                    |
| 5                            | write_contacts                     | Access contacts and profiles                                                      |
| 6                            | get_accounts                       |                                                                                    |
| 7                            | access_fine_location               | The network provider can access both, but the GPS provider can access the fine location only |
| 8                            | access_coarse_location             |                                                                                    |
| 9                            | record_audio                       | Access microphone                                                                 |
| 10                           | read_phone_state                   | Access related telephony features with accessing the phone number, answer calls, track phone status, etc. |
| 11                           | read_phone_numbers                 |                                                                                    |
| 12                           | call_phone                         |                                                                                    |
| 13                           | answer_phone_calls                 |                                                                                    |
| 14                           | read_call_log                      | Access related telephony features with the read and write call log                 |
| 15                           | write_call_log                     |                                                                                    |
| 16                           | process_outgoing_calls             |                                                                                    |
| 17                           | add_voicemail                      |                                                                                    |
| 18                           | use_sip                            | Session initiation protocol                                                       |
| 19                           | body_sensors                       | Access the body or environment sensor                                             |
| 20                           | send_sms                           | Access the message body and send messages                                          |
| 21                           | receive_sms                        |                                                                                    |
| 22                           | read_sms                           |                                                                                    |
| 23                           | receive_wap_push                   |                                                                                    |
| 24                           | receive_mms                        |                                                                                    |
| 25                           | read_external_storage              | Access to external storage                                                        |
| 26                           | write_external_storage             |                                                                                    |

3.3.2. Association of Android Permission and PII

This section used both the heuristic method and a real-life user survey on the available data to correlate between the dangerous permission (PPII) and the PII.

The heuristic method for Android application permission and PII mapping:

The heuristic systemic model (HSM) [84] has already been well practiced while classification reasoning. This study heuristically selected eight PII, those were highly engaged with the Android app’s permissions. While doing this heuristic modeling of PII, the authors used already available knowledge base and literature [46,73,82]. However, this study used a practical assumption to detect eight risky PII (Table 5) where modeling was logical and rational rather than evidence centric. Therefore, to confirm the HSM, this study underwent another level of validation by conducting a survey. Table 5 reflects the final set of PII that are threatened due to aggregated app permissions.

Table 5. Personally identifiable information (PII) class.

| Contact Number | Biometric ID | Address | Social Graph | Human Behavior | Unique ID | E-mail | Location |
|----------------|--------------|---------|--------------|----------------|-----------|--------|----------|

Survey for Android application permission and PII mapping:

With the help of our web app ‘Privacy analysis on mobile App’ (http://52.79.237.144:3000/) [45], a survey was done among 100 students of the computer engineering department, INJE University, Korea on 1 December 2018. The participants often used Kakao and Naver apps for their day to day life.
Before the survey, participants undertook a 3 h long workshop where PII, PPII and Android permission gathering systems and privacy risk were taught. The survey result provided around 2000 unique way of PII leaking through the dangerous Android permissions set. Among those, only 720 unique permissions providing sets were chosen for this study. Figures 5 and 6 provide a few screenshots of the web app [45] where several functions of our proposed web app were visualized. Figure 5 shows the overall state of the ‘Privacy Analysis on Mobile App’ web application followed by Figure 6a,b, which shows detail functions of the application with a closer view.

Figure 5. ‘Privacy analysis on mobile App’ web application (all the apps were owned by Kakao).

Figure 6. ‘Privacy analysis on mobile App’ web application. (a) Aggregated app and permission overview and (b) application detail.

To classify PII, this study only correlated the ‘dangerous permissions’ (Table 4) [59] and PII (Table 5). A brief description of the selected PIIs is illustrated here.

- **Contact Number**: Contact number includes access to the land phone, fax, mobile phone, workplace phone, family member’s phone number, etc.
- **Biometric ID**: Biometric ID includes access to the iris, retina, blood, gene, fingerprint, etc.
- **Address**: Address includes access to the current, home, work, educational institute, frequently covered addresses, etc.
• Social Graph: Social graph includes access to the information of a close friend, neighborhood, cousins, family member, club member, etc.
• Human Behavior: Human behavior includes access to sensitive personal attitude, hobby, buying hobby, brand preference, shopping habit, etc.
• Unique ID: Unique ID includes access to any internet identity, password, membership id, verification code, etc.
• Email: Email includes access to the email domain, email serving company, email address, email-based service preference, email associated photos, etc.
• Location: Finally, the location includes access to the exact location, fine location, job location, recently visited place, family location, workplace, current position, location-based habit data, etc.

An example of the survey output for a student ‘X’ is given in Table 6. After selecting all the used apps from Kakao (owner), permissions like camera, record_audio, body_sensors, read_calendar, access_network, call_phone, send_sms, get_account, read_call_log, etc. were provided to it. Then, the rightmost column, PII class was filled by the user choice. Therefore, based on user ‘X’’s opinion (1) biometric ID was given with legal consent as a result of permitting camera, record_audio, body_sensors and use_fingerprint; (2) location was given with legal consent as a result of permitting read_ext_storage, read_calendar, access_network and access_coarse_location and (3) contact number was given with legal consent as a result of permitting call_phone, send_sms, get_account and read_call_log. The dataset used by this study contained similar information like Table 6.

Table 6. Dangerous Android permission versus personally identifiable information (PII) correlation.

| Case | Dangerous Android Permissions (1 to 26) Owned by a Company | PII Class       |
|------|----------------------------------------------------------|----------------|
| 1    | camera, record_audio, body_sensors, use_fingerprint     | biometric_id   |
| 2    | read_ext_storage, read_calendar, access_network, access_coarse_location | location       |
| 3    | call_phone, send_sms, get_account, read_call_log        | contact number |

4. Case Studies

This section empirically validated the proposed risk model with the case studies. To analyze the large-scale data aggregation, all the app permissions owned by the two owners were considered. For an efficient analysis and visualization, this study presented evaluation results separately by two case studies [85]. Section 4.1 elaborated the experimental environment followed by the two case studies on Kakao (Section 4.2) and Naver (Section 4.3).

4.1. Experimental Environment

4.1.1. Case Study Selection Criteria

The work studied Kakao [43] and Naver [44], the top two South Korean Android app owners (publishers) on the following factors:

1. The number of app users: Both Kakao and Naver own a significant number of users. For instance, ‘KakaoTalk’ application has 220 million registered and 50 million active users [86]. Similarly, 94% of the local search market is owned by the ‘Naver search’ application [87].
2. Application brand popularity in the local market: Both Kakao and Naver are the top application brand in South Korea. Out of the top 10 popular apps, three is from Kakao and two is from Naver [88].
3. The similarity in service: Availability of diverse apps such as chatting, map, dictionary, translator, social, etc. from both the types are another reason for selecting Kakao and Naver.
4. Availability of multiple publishers: Both Kakao and Naver publish apps via several publishers.
4.1.2. Android Privacy Permission Analyzing Web Application

During this work, a web application tool was developed for analyzing and visualizing the risk model. ‘Privacy analysis on mobile app’ (http://52.79.237.144:3000/) [45] was built for analyzing large-scale privacy permission aggregation risk. The proposed app consisted of three parts.

- A node.js based web application server (WAS): The application gathered data from the ‘Google Play-store’ [57] using the ‘open API’. An asynchronous input and output (I/O) based service providing mechanism was implemented using Node.js to provide a non-blocking facility. Regardless of the response success rate of the used API, the proposed application could perform independently.
- A complete database: The database included 18,500 mobile applications with 160,000 dangerous Android permissions. The information of each mobile application included various related information such as title, publisher, downloads and rating of an individual mobile application. It also shows the permission list for each mobile application. Our web application used the Google Play-store [57] API to integrate applications of the Android Play store and our database. In addition, if any application was not in our database that could be added immediately for future use.
- The last part was the web-based service portal: It provided functions like searching, listing and grouping of data in terms of owning a company. The application also allowed users to download an excel file containing the Android permission set and associated app owner’s identity.

4.1.3. Android Privacy Permission Collection

This study listed all the available subsidiaries of Kakao and Naver. Afterward, a previously stated web application was used to download applications, permissions and associated publishers of all the listed subsidiaries of Kakao [43] and Naver [44] for further study. At this point, each entry of the excel sheet contained the owner, publisher, app and associated dangerous permissions. This study combined all the permissions with respect to the publishers and owners. Figure 7 shows two examples of an extensive collection of dangerous permissions (top half of the figure) by applications (bottom half of the figure). Figure 7a,b shows ‘Kakao games corp’ and ‘Daum corp’ of Kakao as an example [37].

![Applications require permissions (owner: Kakao). (a) Kakao Games Corp. and (b) Daum Corp.](image)

4.1.4. Description of the Evaluation Tool

This study used weka [89], a machine learning implementing tool for training and testing dataset. Performance matrices used against classifiers were random forest [90], decision tree [91], PART and bagging cart [92,93]. For evaluating the empirical results four key features like accuracy, precision,
recall and F1 score \cite{94} were used. One of the key trials for measuring whether a proposed classification is true or not is accuracy. It indicates the amount of true positive and true negative. The closer the accuracy is to 1, the better the used classification is performed. Precision refers to the degree to which repeated measurements under the same conditions show an identical outcome. The percentage of total relevant results correctly classified by the algorithm is known as recall. The recall is the number of real positives that are identified accurately. As precision and recall come at the cost of another, it is highly unlikely to maximize both the metrics simultaneously. Consequently, another significant metric named F-1 score originates.

4.2. Case Study of Kakao

4.2.1. Description of the Dataset

As the total number of instances for Kakao dataset was 326, this study used 85% (277) of randomly chosen instances for training purposes and the rest (15%; 49) was used for testing. The study used 26 dangerous permissions and eight PII classes for the Kakao case study (Table 7).

Table 7. Numerical overview of the dataset used for the Kakao case study.

| Owners       | Publishers      | Number of Apps | Possible Permission Set (Rows) | Considered PPII (Dangerous Android Permissions) | Considered PII Class |
|--------------|-----------------|----------------|-------------------------------|-----------------------------------------------|----------------------|
| Kakao [43]   | Kakao corporation | 20             |                               | 326                                           | 26                   |
|              | Kakao mobility   | 8              |                               |                                               |                      |
|              | Kakao game corp  | 16             |                               |                                               |                      |
|              | Kakao theme      | 11             |                               |                                               |                      |
|              | Daum corporation | 8              |                               |                                               |                      |

4.2.2. Key Insight from Experimental Evaluation

Figure 8 shows the performance measuring matrices such as the accuracy, precession, recall and f1 score against the key machine learning classification algorithm like the random forest, decision tree, PART and bagging cart. The factors and classification method used in this study are globally recognized \cite{94}. Figure 8 shows, for the Kakao [43] dataset, decision tree achieved a better outcome than the other classifiers in terms of accuracy. Maximum accuracy achieved while classifying PII from dangerous Android permission was 71%. For the Kakao case, the decision tree-based classifier achieved a maximum precision of 0.76. This study achieved 0.70 recall for decision tree. Finally, the F1 score (F measure) was recorded as 0.69. Figure 8 shows the performance record for Kakao tested by the study.
4.2. Case Study of Kakao

4.2.1. Description of the Dataset

As the total number of instances for the Kakao dataset was 326, this study used 85% (277) of randomly chosen instances for training purposes and the rest (15%; 49) was used for testing. The study used 26 dangerous permissions and eight PII classes for the Kakao case study (Table 7).

Table 7. Numerical overview of the dataset used for the Kakao case study.

| Owners    | Publishers          | Number of Apps | Possible Permission Set (Rows) | Considered PPII (Dangerous Android Permissions) | Considered PII Class |
|-----------|---------------------|----------------|-------------------------------|-----------------------------------------------|----------------------|
| Kakao     | corporation         | 20             | 326                           | 26                                            | 8                    |
| Kakao     | mobility            | 8              |                               |                                               |                      |
| Kakao     | game corp           | 16             |                               |                                               |                      |
| Daum      | corporation         | 8              |                               |                                               |                      |

4.2.2. Key Insight from Experimental Evaluation

Figure 8 shows the performance measuring matrices such as the accuracy, precision, recall and f1 score against the key machine learning classification algorithm like the random forest, decision tree, PART and bagging cart. The factors and classification method used in this study are globally recognized [94]. Figure 8 shows, for the Kakao [43] dataset, decision tree achieved a better outcome than the other classifiers in terms of accuracy. Maximum accuracy achieved while classifying PII from dangerous Android permission was 71%. For the Kakao case, the decision tree-based classifier achieved a maximum precision of 0.76. This study achieved 0.70 recall for decision tree. Finally, the F1 score (F measure) was recorded as 0.69. Figure 8 shows the performance record for Kakao tested by the study.

Figure 8. Evaluation result on Kakao.

4.3. Case Study of Naver

4.3.1. Description of the Dataset

As the total number of instances for the Naver dataset was 394, this study used 85% (335) of randomly chosen instances for training and the rest (15%; 60) was used for testing. The study used 26 dangerous permissions and eight PII classes for the Naver case study (Table 8).

Table 8. Numerical overview of the dataset used for the Naver case study.

| Owners     | Publishers       | Number of Apps | Possible Permission Set (Rows) | Considered PPII (Dangerous Android Permissions) | Considered PII Class |
|------------|------------------|----------------|-------------------------------|-----------------------------------------------|----------------------|
| Naver [44] | corporation      | 36             | 394                           | 26                                            | 8                    |
|            | Line corporation | 19             |                               |                                               |                      |

4.3.2. Key Insight from Experimental Evaluation

Figure 9 shows the performance measuring matrices such as the accuracy, precision, recall and f1 score against the globally recognized machine learning classification algorithm like the random forest, decision tree, PART and bagging cart [94]. Figure 9 shows a maximum of 75% accuracy was achieved with the Naver [44] data while classifying PII by the random forest classification. Maximum precision was recorded as 0.81. This study achieved 0.75 recall for the random forest. The stated F1 score by this work for Naver was 0.74. Figure 9 shows detail PII classification performance record for Naver.
4.3. Case Study of Naver

4.3.1. Description of the Dataset

As the total number of instances for the Naver dataset was 394, this study used 85% (335) of randomly chosen instances for training and the rest (15%; 60) was used for testing. The study used 26 dangerous permissions and eight PII classes for the Naver case study (Table 8).

Table 8. Numerical overview of the dataset used for the Naver case study.

| Owners     | Publishers   | Number of Apps | Possible Permission Set (Rows) | Considered PPI (Dangerous Android Permissions) | Considered PII Class |
|------------|--------------|----------------|-------------------------------|-----------------------------------------------|---------------------|
| Naver      | Naver corporation | 36             | 394                           | 26                                            | 8                   |
| Line       | Line corporation | 19             |                               |                                               |                     |

4.3.2. Key Insight from Experimental Evaluation

Figure 9 shows the performance measuring matrices such as the accuracy, precision, recall and f1 score against the globally recognized machine learning classification algorithm like the random forest, decision tree, PART and bagging cart. Figure 9 shows a maximum of 75% accuracy was achieved with the Naver data while classifying PII by the random forest classification. Maximum precision was recorded as 0.81. This study achieved 0.75 recall for the random forest. The stated F1 score by this work for Naver was 0.74. Figure 9 shows detail PII classification performance record for Naver.

5. Discussions

5.1. Key Learning From the Study

Based on the survey results and empirical evidence, the study highlighted the possible trends of permission aggregation.

- On the severity of access, Android permissions were of two types of one-time access and continuous access. PII like ‘biometric IDs’ and ‘contact numbers’ were directly exposed through one-time access of ‘USE_FINGERPRINT’ and ‘CELL_PHONE’. Alternatively, ‘social graph’ and ‘human behavior’ required continuous access to ‘READ_CALL_LOG’, ‘READ_EXT_STORAGE’, ‘READ_CALENDAR’, etc.

- PII exposing risk depended on two sub-factors like quantitative and qualitative attributes of the Android permissions. For example, PII like ‘location’ (location graph) got more accurate after reiteration of ‘FINE_LOCATION’ and ‘COURSE_LOCATION’ permission. That is, the more specific the Android permissions used, the more the PII is exposed. Alternatively, the permission ‘BODY_SENSOR’ sensed surroundings, preferably the human body continuously. However, few ‘BODY_SENSOR’s might collect a high-quality attribute, some might not get to. Thus, the user profiling risk (PII exposing rate) also depended on the data quality gathered through permissions.

- The PII revealing risk did not depend on a single Android app (permission). As the publishers and owners could aggregate permissions from different apps, awareness and research on a single Android application (permission) were not enough anymore. Due to owner side permission accumulation risk, risk modeling studies must evaluate from multiple app perspectives.

5.2. Probabilistic Analysis of the Risk Model

The proposed study was a novel approach that linked the leakage of PII with aggregated Android permissions. Although a quantitative perspective or the accuracy of the study was limited, consideration of permission aggregation at the manufacturer side increased the qualitative value of
the risk model. So far, no other study has considered how heavyweight organizations can gather PII through Android applications, more specifically, through aggregated Android permissions.

The probabilistic analysis of the risk model was discussed in two phases. Firstly, permission aggregation by publishers from multiple underneath applications. The likelihood of breaching individual PII via app permission App1 to Appn towards Pub1 to Pubn was \( P_1(A), P_2(B), P_3(C) \ldots P_n(N) \). Similarly, the likelihood of getting a specific PII for aggregated app permissions was distributed as \( P_a(A) \cup P_b(B) \cup P_c(C) \). Since individual PII is mutually exclusive in relation to Android permissions, the same privacy permission may be responsible for several PII leaking. The chance of a single PII leakage to a specific publisher from the list of publishers Pub1 to Pubn is independent, that is \( P_{a \to n} (A \text{ and } B \text{ and } C \ldots N) = 0 \). Secondly, the study proposed the chance of getting eight PII classes (contact number, biometric ID, address, social graph, human behavior, unique ID, email and location) from the combination of a specific permission set. To discuss the probability of a particular PII class leakage from android permission, we would first have a brief look at probability estimation of decision tree and random forest from the knowledge of existing literature [95,96]. The class likelihoods for an end node were projected by the comparative occurrence of that group in the very last node (Figure 10).

![Figure 10. Probability of revealing a PII.](image)

The probability estimation of the tree for a new class is the likelihood of a particular class linked with the terminal node \((\text{PII}_{1 \to 4})\) Figure 10. The result estimation for the random forest was done by the aggregated result of individual probability of a new subject among all the trees. Actually, each and every tree forming the random forest classification algorithm was translated as a logistic regression probabilistic model. Figure 10 illustrates the aforementioned single-tree representation concept in detail. For a terminal node or class \((\text{PII}_{1 \to 4})\) all the events to get to that terminal or class were considered. For example, the conditional probability for \(\text{PII}_4\) could be estimated as \( P(y = 1|x_1 \leq p_1, x_2 \leq p_2, x_3 \leq p_3) \), Where, \( y \) is the dichotomous outcome of the probability of leaking a particular class. Similarly, \( p_1, p_2 \) and \( p_3 \) were the points where the tree had been spat for decision-making. The probability of leaking each class from \(\text{PII}_{1 \to 4}\) is defined as:

\[
\text{PII}_1 = \begin{cases} 
1, & x_1 > p_1 \\
0, & \text{else} 
\end{cases}
\]

\[
\text{PII}_2 = \begin{cases} 
1, & x_2 > p_2, x_1 \leq p_1 \\
0, & \text{else} 
\end{cases}
\]

\[
\text{PII}_3 = \begin{cases} 
1, & x_3 > p_3, x_2 \leq p_2, x_1 \leq p_1 \\
0, & \text{else} 
\end{cases}
\]

\[
\text{PII}_4 = \begin{cases} 
1, & x_4 > p_4 \\
0, & \text{else} 
\end{cases}
\]
\[ PII_4 = \begin{cases} 1, & x_3 \leq p_3, x_2 \leq p_2, x_1 \leq p_1 \\ 0, & \text{else} \end{cases} \]

If all the trees that consist of random forest are translated like the above procedure followed by the re-estimation of probability for the individual tree, the probability of the proposed PII classification can be calculated. Similarly, relative class frequency (RF) and Laplace estimation can also estimate the probability estimation tree (PET) [95].

5.3. Recommendations

The study made the following recommendations for Android application users:

- Along with the assessment of an application privacy features, publisher’s and owner’s identity must be noted. The study recommended using applications from different publishers and owners.
- The study also recommended the users to track the total flow of dangerous Android permissions to any specific application publishers or owners. For example, the user should seriously monitor which of the sensitive PII is about to be revealed to a specific publisher or owner.
- Most of the studies were dealing with Android permissions to personal data leak classification by considering the first level (application level) information. However, this study recommended researchers to consider the second (publishers) and third level (owner) data (permissions) aggregation. Otherwise, the overall research outcomes and objective might be missed.

5.4. Limitations

As the main goal of the risk model is to consider the permission aggregation caused privacy risk to identical owners, the study might have limited accuracy. The accuracy of our proposed classification showed a lower score in comparison to other Android application based PII classification approaches [68,97]. The size of the dataset was one of the major limitations of this study. The study was an ongoing work where only two owners were considered by the case studies. During the machine learning algorithm evaluation, we could train and test our proposal with a limited amount of data. Naturally, 5–10 publishers from owner publish apps in the Play store [57]. The permission versus PII dataset with manufacturer tagging had to be done manually. This study might be affected by a lack of consideration of a weighting factor of individual Android permissions, as this study assumed individual permission carried equal attributes (data significance). Similarly, proposed data aggregation-based classification overlooked the effect of race, gender, age, country, etc. minor factors while classifying the PII threat. Therefore, the PII might have classified slightly differently based on the aforesaid factors. Finally, the quality of the analyzed data features was reasonably good considered in the case study; however, the number of considered features were small as only dangerous permissions identified were considered. Therefore, in future research, every Android permission will be measured.

5.5. Future Consideration

Our future study plan is to consider the followings to make the work efficient and user-friendly.

- Considering the diverse data sharing scope among stakeholder: Data sharing scope among stakeholders (owners, publishers and applications) is variable. In future, the study plans to consider a distinct data sharing likelihood from the application to the publisher (α), the publisher to the owner (β) and the owner to owner (γ). The inter-company relationship and data-storing architecture decide the data sharing options. Permission data acquisition rate would vary based on the aforesaid factor.
- The weighting of the individual Android permission: Not all Android permission carries equal information that can form PII. The study plans to introduce a data value weight (ω₁, ω₂...ωₙ) on each Android dangerous permission based on its contribution of PII formation. The dangerous Android permission to PII formation matrices will be redefined based on these weights.
• Redesigning the analyzing app: Finally, the study plans to improve the web app ‘Privacy analysis on mobile App’ (http://52.79.237.144:3000) [45] based on the aforesaid factors. The web application will be reshaped in such a way where the user can track the percentage of the overall personal data flow through Android permissions towards a specific owner.
• In the future, a larger dataset with higher application owning enterprises, such as Alphabet, Facebook, eBay, Amazon, etc. would be considered

6. Conclusions

This study presented a novel personal information risk classification model that was inspired by the permission aggregation concept. Android permissions congregated through the mobile applications, publishers and owners were considered by the risk model. Different classifiers were used and compared to classify vulnerable PII. A web application (http://52.79.237.144:3000) [45] was developed during the study in order to visualize the actual fact of large-scale privacy permission aggregation. With the latest Google-Play API, our web application [45] provided users with a better understanding of the large-scale Android permissions aggregation by influential owners.

The study illustrated how permission aggregation empowered entrepreneurs with a large amount of partial information. Consequently, this partial information amplifies the chance of user identity re-identification (user profiling). While classifying the leakage of specific PII from a set of Android permissions, the study considered the top two South Korean app owners (Kakao and Naver) along with their publishers and their published apps. The study achieved a satisfactory accuracy by adopting a decision tree and random forest classification algorithm to validate the data aggregation caused PII risk model. As data accumulation has become a key business imperative that challenges several user identities, the study concluded the contact number, biometric ID, address, social graph, human behavior, unique ID, email and location information as frequently exposed PII.

The analytical results demonstrated that the classification approach was guaranteed to find a higher rate of PII risk towards large enterprises. The proposed risk model concept could also be used for marketing policy-making for the data-driven economy. In the future, the study plans to develop a privacy-aware permission sharing and tracking tool that can check the real-time permission movement towards the decisive owners. In addition, the study plans to apply the proposed risk model for the IoT domain as well as most of the IoT devices are managed by the Android application.

Author Contributions: Conceptualization, M.M.H.O. and J.Y.; Formal analysis, M.M.H.O.; Funding acquisition, C.-S.K.; Investigation, N.-Y.L.; Methodology, M.M.H.O.; Project administration, C.-S.K.; Resources, J.Y.; Software, J.Y.; Supervision, C.-S.K.; Validation, J.Y.; Visualization, N.-Y.L.; Writing—original draft, M.M.H.O.; Writing—review & editing, N.-Y.L.

Funding: This work was funded by the Institute of Information and Communications Technology Planning and Evaluation (IITP), Ministry of Science and ICT, Korea. Grant No. 2018–0-00261, “GDPR Compliant Personally Identifiable Information Management Technology for IoT Environment “ and “The APC was funded by the Institute of Information and Communications Technology Planning and Evaluation (IITP), Ministry of Science and ICT, Korea. Grant No. 2018–0-00261, “GDPR Compliant Personally Identifiable Information Management Technology for IoT Environment”.

Acknowledgments: The authors thanks Kim Dong Yong of KAIST for helping in developing the web application. The authors also acknowledge Nuzhat Mariam for language proofreading and revision.

Conflicts of Interest: The authors declare no conflict of interest.

References
1. Chong, I.; Ge, H.; Li, N.; Proctor, R.W. Influence of privacy priming and security framing on mobile app selection. Comput. Secur. 2018, 78, 143–154. [CrossRef]
2. Onik, M.M.H.; Ahmed, M. Blockchain in the Era of Industry 4.0. In Data Analytics: Concepts, Techniques, and Applications; Mohiuddin Ahmed, A.-S.K.P., Ed.; CRC Press: Boca Raton, FL, USA, 2018; pp. 259–298, ISBN 9781138500815.
3. Ahmed, E.; Yaqoob, I.; Hashem, I.A.T.; Shuja, J.; Imran, M.; Guizani, N.; Bakhsh, S.T. Recent Advances and Challenges in Mobile Big Data. *IEEE Commun. Mag.* 2018, 56, 102–108. [CrossRef]
4. Cadwalladr, C.; Graham-Harrison, E. Revealed: 50 Million Facebook Profiles Harvested for Cambridge Analytica in Major Data Breach. Available online: https://www.theguardian.com/news/2018/mar/17/cambridge-analytica-facebook-influence-us-election (accessed on 12 February 2019).
5. Volodzko, D. Marriott Breach. Available online: https://www.forbes.com/sites/davidvolodzko/2018/12/04/marriott-breach-exposes-far-more-than-just-data/#19e9b70f6297 (accessed on 12 February 2019).
6. Kenthapadi, K.; Mironov, I.; Thakurta, A. Privacy-preserving Data Mining in Industry. In Proceedings of the 2019 World Wide Web Conference, Taipei, Taiwan, 20–25 April 2019; pp. 1308–1310.
7. He, Y.; Yang, X.; Hu, B.; Wang, W. Dynamic privacy leakage analysis of Android third-party libraries. *J. Inf. Secur. Appl.* 2019, 46, 259–270. [CrossRef]
8. Jha, A.K.; Lee, W.J. An empirical study of collaborative model and its security risk in Android. *J. Syst. Softw.* 2018, 137, 550–562. [CrossRef]
9. Yu, L.; Luo, X.; Qian, C.; Wang, S.; Leung, H.K.N. Enhancing the Description-to-Behavior Fidelity in Android Apps with Privacy Policy. *IEEE Trans. Softw. Eng.* 2018, 44, 834–854. [CrossRef]
10. Ito, K.; Hasegawa, H.; Yamaguchi, Y.; Shimada, H.Y. Detecting Privacy Information Abuse by Android Apps from API Call Logs. In Proceedings of the 2018 International Workshop on Security, Miyagi, Japan, 3–5 September 2018; pp. 143–157.
11. Islam, M.R. Numeric rating of Apps on Google Play Store by sentiment analysis on user reviews. In *Proceedings of the 2014 International Conference on Electrical Engineering and Information & Communication Technology*; IEEE: Piscataway, NJ, USA, 2014; pp. 1–4.
12. Hatamian, M.; Momen, N.; Fritsch, L.; Rannenberg, K. A Multilateral Privacy Impact Analysis Method for Android Apps. In *Annual Privacy Forum*; Springer: Cham, UK, 2019; pp. 87–106.
13. Azfar, A.; Choo, K.-K.R.; Liu, L. An Android Communication App Forensic Taxonomy. *J. Forensic Sci.* 2016, 61, 1337–1350. [CrossRef] [PubMed]
14. Azfar, A.; Choo, K.-K.R.; Liu, L. Forensic Taxonomy of Android Social Apps. *J. Forensic Sci.* 2017, 62, 435–456. [CrossRef]
15. Mehrnezhad, M.; Toreini, E. What Is This Sensor and Does This App Need Access to It? *Informatics* 2019, 6, 7. [CrossRef]
16. Gu, J.; Huang, R.; Jiang, L.; Qiao, G.; Du, X.; Guizani, M. A Fog Computing Solution for Context-Based Privacy Leakage Detection for Android Healthcare Devices. *Sensors* 2019, 19, 1184. [CrossRef] [PubMed]
17. Moore, S.R.; Ge, H.; Li, N.; Proctor, R.W. Cybersecurity for Android Applications: Permissions in Android 5 and 6. *Int. J. Hum. Comput. Interact.* 2019, 35, 630–640. [CrossRef]
18. McLroy, S.; Shang, W.; Ali, N.; Hassan, A.E. User reviews of top mobile apps in Apple and Google app stores. *Commun. ACM* 2017, 60, 62–67. [CrossRef]
19. Wang, X.; Wang, W.; He, Y.; Liu, J.; Han, Z.; Zhang, X. Characterizing android apps’ behavior for effective detection of malapps at large scale. *Futur. Gener. Comput. Syst.* 2017, 75, 30–45. [CrossRef]
20. Kumar, R.; Zhang, X.; Khan, R.U.; Sharifi, A. Research on Data Mining of Permission-Induced Risk for Android IoT Devices. *Appl. Sci.* 2019, 9, 277. [CrossRef]
21. Kim, J.; Jung, I. Efficient Protection of Android Applications through User Authentication Using Peripheral Devices. *Sustainability* 2018, 10, 1290. [CrossRef]
22. Liu, X.; Du, X.; Zhang, X.; Zhu, Q.; Wang, H.; Guizani, M. Adversarial Samples on Android Malware Detection Systems for IoT Systems. *Sensors* 2019, 19, 974. [CrossRef] [PubMed]
23. Doğrulu, L.; KIRAZ, O. Web-based android malicious software detection and classification system. *Appl. Sci.* 2018, 8, 1622. [CrossRef]
24. Duffie, D.; Malamud, S.; Manso, G. The relative contributions of private information sharing and public information releases to information aggregation. *J. Econ. Theory* 2010, 145, 1574–1601. [CrossRef]
25. Richter, H.; Slowinski, P.R. The Data Sharing Economy: On the Emergence of New Intermediaries. *IIC-Int. Rev. Intellect. Prop. Compet. Law* 2019, 50, 4–29. [CrossRef]
26. Venkatadri, G.; Luchereini, E.; Sapieyevski, P.; Mislove, A. Investigating sources of PII used in Facebook’s targeted advertising. *Proc. Priv. Enhancing Technol.* 2019, 2019, 227–244. [CrossRef]
27. Huckvale, K.; Torous, J.; Larsen, M.E. Assessment of the data sharing and privacy practices of smartphone apps for depression and smoking cessation. *JAMA Netw. Open* 2019, 2, 192542. [CrossRef]
28. Shilton, K.; Greene, D. Linking Platforms, Practices, and Developer Ethics: Levers for Privacy Discourse in Mobile Application Development. J. Bus. Ethics 2019, 155, 131–146. [CrossRef]
29. Facebook, Instagram, WhatsApp Go down Simultaneously. Available online: https://www.businesstoday.in/technology/internet/facebook-instagram-whatsapp-go-down-simultaneously/story/327610.html (accessed on 6 August 2019).
30. Facebook is Sharing Users’ WhatsApp and Instagram Data to Catch Terrorists | The Independent. Available online: https://www.independent.co.uk/life-style/gadgets-and-tech/news/facebook-policy-share-users-data-across-whatsapp-instagram-tackle-terrorists-social-media-app-isis-a7797201.html (accessed on 29 July 2019).
31. Rangole, W.F.H.K. Large-Scale Authorization Data Collection and Aggregation. US Patents 16/056,322, 7 March 2019.
32. Boehm, M.; Evfimievski, A.; Reinwald, B. Efficient data-parallel cumulative aggregates for large-scale machine learning. In BTW 2019; Grust, T., Naumann, F., Böhm, A., Lehner, W., Hälder, T., Rahm, E., Heuer, A., Klettke, M., Meyer, H., Eds.; Gesellschaft für Informatik: Bonn, Germany, 2019; pp. 267–286.
33. Bakalash, R.; Shaked, G.; Caspi, J. Enterprise-Wide Data-Warehouse with Integrated Data Aggregation Eng. U.S. Patent 7,315,849, 1 January 2008.
34. Rabl, T.; Gómez-Villamor, S.; Sadoghi, M.; Muntés-Mulero, V.; Jacobsen, H.A.; Mankovskii, S. Solving big data challenges for enterprise application performance management. Proc. VLDB Endow. 2012, 5, 1724–1735. [CrossRef]
35. Fritsch, L.; Momen, N. Derived Partial Identities Generated from App Permissions. In Open Identity Summit 2017; Fritsch, L., Roßnagel, H., Hühnlein, D., Eds.; Gesellschaft für Informatik: Bonn, Germany, 2017; pp. 117–130.
36. Jinhong, Y.; Chul-Soo, K.I.M.; ONIK, M.M.H. Aggregated Risk Modelling of Personal Data Privacy in Internet of Things. In Proceedings of the 2019 21st International Conference on Advanced Communication Technology (ICACT); IEEE: Piscataway, NJ, USA, 2019; pp. 425–430.
37. Minen, M.T.; Stieglitz, E.J.; Sciortino, R.; Torous, J. Privacy Issues in Smartphone Applications: An Analysis of Headache/Migraine Applications. Headache J. Head Face Pain 2018, 58, 1014–1027. [CrossRef] [PubMed]
38. Gündüç, S.; Eryiğit, R. Role of new ideas in the mobile phone market share. Int. J. Model. Simul. Sci. Comput. 2018, 9, 1850018. [CrossRef]
39. Kakao. Available online: https://www.kakaocorp.com/ (accessed on 6 April 2019).
40. Naver. Available online: https://www.naver.com/ (accessed on 6 April 2019).
41. Privacy Analysis on Mobile App. Available online: http://52.79.237.144;3000/ (accessed on 11 May 2019).
42. Voss, W.G. European union data privacy law reform: General data protection regulation, privacy shield, and the right to delisting. Bus. Lawyer 2017, 72, 221–223.
43. McCallister, E. Guide to Protecting the Confidentiality of Personally Identifiable Information; Diane Publishing: Collingdale, PA, USA, 2010; ISBN 1437934889.
44. Pfitzmann, A.; Hansen, M. Anonymity, unlinkability, undetectability, unobservability, pseudonymity, and identity management—a consolidated proposal for terminology. Version v0 2008, 31, 15.
50. Kaur, G.; Agrawal, S. Differential Privacy Framework: Impact of Quasi-identifiers on Anonymization. In Proceedings of the 2nd International Conference on Communication, Computing and Networking, Chandigarh, India, 29–30 March 2018; pp. 35–42.

51. Soh, C.; Njilla, L.; Kwiat, K.; Kamhoua, C.A. Learning quasi-identifiers for privacy-preserving exchanges: A rough set theory approach. *Granul. Comput.* 2018, 3, 1–14.

52. Murphy, R.S. Property rights in personal information: An economic defense of privacy. In *Privacy*; Routledge: London, UK, 2017; pp. 43–79.

53. NIST PII. Available online: [https://csrc.nist.gov/glossary/term/personally-identifiable-information](https://csrc.nist.gov/glossary/term/personally-identifiable-information) (accessed on 12 February 2019).

54. Butler, D.A.; Rodrick, S. *Australian Media Law*; Thomson Reuters (Professional) Australia Limited: Pyrmont, NSW, Australia, 2015; ISBN 9780455234403.

55. Porter, C.C. De-identified data and third party data mining: The risk of re-identification of personal information. *Shidler JL Com. Tech.* 2008, 5, 1.

56. Momen, N.; Pulls, T.; Fritsch, L.; Lindskog, S. How Much Privilege Does an App Need? Investigating Resource Usage of Android Apps (Short Paper). In *Proceedings of the 2017 15th Annual Conference on Privacy, Security and Trust (PST)*; IEEE: Piscataway, NJ, USA, 2017; pp. 268–2685.

57. Android Apps on Google Play. Available online: [https://play.google.com/store/apps](https://play.google.com/store/apps) (accessed on 6 April 2019).

58. iTunes - Apple (IN). Available online: [https://www.apple.com/kr/itunes/](https://www.apple.com/kr/itunes/) (accessed on 1 July 2019).

59. Google Developers Android Dangerous Permissions. Available online: [https://developer.android.com/guide/topics/permissions/overview](https://developer.android.com/guide/topics/permissions/overview) (accessed on 12 December 2018).

60. Sharma, U.; Bansal, D. A Study of Android Application Execution Trends and Their Privacy Threats to a User with Respect to Data Leaks and Compromise. In *Advanced Computational and Communication Paradigms*; Springer: Singapore, 2018; pp. 675–682, ISBN 978-981-10-8237-5.

61. Baalous, R.; Poet, R. How Dangerous Permissions are Described in Android Apps’ Privacy Policies? In *Proceedings of the 11th International Conference on Security of Information and Networks*, Glasgow, UK, 9–11 September 2018; pp. 26–27.

62. Sivan, N.; Bitton, R.; Shabtai, A. Analysis of Location Data Leakage in the Internet Traffic of Android-based Mobile Devices. *arXiv* 2018, arXiv:1812.04829.

63. Kim, K.; Kim, T.; Lee, S.; Kim, S.; Kim, H. When Harry Met Tinder: Security Analysis of Dating Apps on Android. In *Proceedings of the 2018 Nordic Conference on Secure IT Systems*, Oslo, Norway, 28–30 November 2018; pp. 454–467.

64. Onik, M.M.H.; Al-Zaben, N.; Hoo, H.P.; Kim, C.-S. A Novel Approach for Network Attack Classification based on Sequential Questions. *Ann. Emerg. Technol. Comput.* 2018, 2, 1–14. [CrossRef]

65. Chiluka, N.; Singh, A.K.; Eswarawaka, R. Privacy and Security Issues Due to Permissions Glut in Android System. In *Proceedings of the 2018 International Conference on Inventive Research in Computing Applications (ICIRCA)*, Coimbatore, India, 11–12 July 2018; pp. 406–411.

66. Jadon, P.; Mishra, D.K. Security and Privacy Issues in Big Data: A Review. In *Emerging Trends in Expert Applications and Security*; Springer: Singapore, 2019; pp. 659–665, ISBN 978-981-13-2285-3.

67. Jain, V.; Laxmi, V.; Gaur, M.; On, M.M. APPLADroid: Automaton Based Inter-app Privacy Leak Analysis for Android. In *Proceedings of the 2019 International Conference on Security & Privacy*, Prague, Czech Republic, 23–25 February 2019; pp. 219–233.

68. Sharma, K.; Gupta, B.B. Towards Privacy Risk Analysis in Android Applications Using Machine Learning Approaches. *Int. J. E Serv. Mob. Appl.* 2019, 11, 1–21. [CrossRef]

69. Song, D.H.; Son, C.Y. Mismanagement of personally identifiable information and the reaction of interested parties to safeguarding privacy in South Korea. *Inf. Res.* 2017, 22, 1–16.

70. Sadeghi, A.; Bagheri, H.; Garcia, J.; Malek, S. A taxonomy and qualitative comparison of program analysis techniques for security assessment of android software. *IEEE Trans. Softw. Eng.* 2017, 43, 492–530. [CrossRef]

71. Li, J.; Sun, L.; Yan, Q.; Li, Z.; Srissa-an, W.; Ye, H. Significant Permission Identification for Machine Learning Based Android Malware Detection. *IEEE Trans. Ind. Inform.* 2018, 14, 3216–3225. [CrossRef]
72. Arora, A.; Peddoju, S.K.; Chouhan, V.; Chaudhary, A. Poster: Hybrid Android Malware Detection by Combining Supervised and Unsupervised Learning. In Proceedings of the 24th Annual International Conference on Mobile Computing and Networking, New Delhi, India, 29 October–2 November 2018; pp. 798–800.

73. Shuba, A.; Bakopoulou, E.; Markopoulou, A. Privacy Leak Classification on Mobile Devices. In Proceedings of the IEEE Workshop on Signal Processing Advances in Wireless Communications, SPAWC, Kalamata, Greece, 25–28 June 2018.

74. Razaghpahnan, A.; Nithyanand, R.; Vallina-Rodriguez, N.; Sundaresan, S.; Allman, M.; Kreibich, C.; Gill, P. Apps, Trackers, Privacy, and Regulators: A Global Study of the Mobile Tracking Ecosystem. In Proceedings of the 2018 Network and Distributed System Security Symposium, San Diego, CA, USA, 18–21 February 2018.

75. Ren, J.; Rao, A.; Lindorfer, M.; Legout, A.; Choffnes, D. Recon: Revealing and controlling pii leaks in mobile network traffic. In Proceedings of the 14th Annual International Conference on Mobile Systems, Applications, and Services, Singapore, 25–30 June 2016; pp. 361–374.

76. Daly, A. The legality of deep packet inspection. *Int. J. Commun. Law Policy* 2011, 14, 1–12. [CrossRef]

77. Cohen, I.G.; Mello, M.M. HIPAA and protecting health information in the 21st Century. *JAMA* 2018, 320, 231–232. [CrossRef]

78. Onik, M.M.H.; Kim, C.S.; Yang, J. Personal Data Privacy Challenges of the Fourth Industrial Revolution. In Proceedings of the 2019 21st International Conference on Advanced Communication Technology (ICACT), PyeongChang, Korea, 17–20 February 2019; pp. 635–638.

79. Permissions Overview. Android Developers. Available online: https://developer.android.com/guide/topics/permissions/overview (accessed on 5 April 2019).

80. Galloway, S. *The Four: The Hidden DNA of Amazon, Apple, Facebook and Google*; Bantam Press: London, UK, 2017; ISBN 978-0525501220.

81. The Data Brokers Quietly Buying and Selling Your Personal Information. Available online: https://www.fastcompany.com/90310803/here-are-the-data-brokers-quietly-buying-and-selling-your-personal-information (accessed on 11 July 2019).

82. Liccardi, I.; Pato, J.; Weitzner, D.; Abelson, H.; De Roure, D. No technical understanding required: Helping users make informed choices about access to their personal data. In Proceedings of the 11th International Conference on Mobile and Ubiquitous Systems: Computing, Networking and Services, Houston, TX, USA, 12–14 November 2014; pp. 140–150.

83. Kumar, S.; Shanker, R. Context Aware Dynamic Permission Model: A Retrospect of Privacy and Security in Android System. In Proceedings of the 2018 International Conference on Intelligent Circuits and Systems, Hsinchu, Taiwan, 18–20 March 2018; pp. 324–329.

84. Todorov, A.; Chaiken, S.; Henderson, M.D. The heuristic-systematic model of social information processing. In *The Persuasion Handbook: Developments in Theory and Practice*; SAGE Publications Sage UK: London, UK, 2002; pp. 195–211.

85. Creswell, J.; Poth, C. *Qualitative Inquiry and Research Design: Choosing among Five Approaches*; Sage Publications Sage CA: Thousand Oaks, CA, USA, 2017.

86. Markov, Z.; Russell, I. An introduction to the WEKA data mining system. *ACM SIGCSE Bull.* 2006, 38, 367–368. [CrossRef]

87. Liaw, A.; Wiener, M. Classification and regression by randomForest. *R news* 2002, 2, 18–22.

88. Breiman, L. Bagging predictors. *Mach. Learn.* 1996, 24, 123–140. [CrossRef]
94. Powers, D.M. Evaluation: From precision, recall and F-measure to ROC, informedness, markedness and correlation. *J. Mach. Learn. Technol.* 2011, 2, 37–63.

95. Boström, H. Estimating Class Probabilities in Random Forests. In Proceedings of the 2007 Sixth International Conference on Machine Learning and Applications, ICMLA 2007, Cancun, Mexico, 13–15 December 2007; pp. 211–216.

96. Dankowski, T.; Ziegler, A. Calibrating random forests for probability estimation. *Stat. Med.* 2016, 35, 3949–3960. [CrossRef]

97. Wei, L.; Luo, W.; Weng, J.; Zhong, Y.; Zhang, X.; Yan, Z. Machine learning-based malicious application detection of android. *IEEE Access* 2017, 5, 25591–25601. [CrossRef]