Revealing the Unrevealed: Mining Smartphone Users Privacy Perception on App Markets

Majid Hatamian[a,*], Jetzabel Serna[a], Kai Rannenberg[a]

[a] *Chair of Mobile Business & Multilateral Security*
Goethe University Frankfurt
Frankfurt am Main, Germany

Abstract

Popular smartphone apps may receive several thousands of user reviews containing statements about apps' functionality, interface, user-friendliness, etc. They sometimes also comprise privacy relevant information that can be extremely helpful for app developers to better understand why users complain about certain privacy aspects of their apps. However, due to the complicated and sometimes vague nature of reviews, it is quite though and time consuming for developers to go through all these reviews to get information about privacy aspects of apps. Furthermore, previous studies confirmed that sometimes bad privacy practices happen due to the app developers' lack of knowledge in API definition and usage. In addition, such information can be useful for mobile users as the lack of privacy indicators in smartphone ecosystems prevents them from being able to compare apps in terms of privacy and to perform informed privacy decision making when selecting apps. Therefore, in this paper we propose *Mobile App Reviews Summarization (MARS)* to overcome the aforementioned difficulties. We exploit user reviews on the Google Play Store as a relevant source in order to extract and quantify privacy relevant claims associated with apps. Based on Machine Learning (ML), Natural Language Processing (NLP) and sentiment analysis techniques, *MARS* detects privacy relevant reviews and categorizes them into a pre-identified list of privacy threats in the context of mobile apps. The combination of these concepts provides developers with specific knowledge about the privacy threats and behavior of apps based on user generated reports that are otherwise difficult to detect. Not only developers, but also users can benefit from such mechanism to compare apps in terms of privacy aspects. To this end, we complement *MARS* by a novel app behavior monitoring tool that further enhances the whole reliability of the results generated by *MARS*. Our results demonstrate the applicability of our approach which provides precision, recall and F-score as high as 94.84%, 91.30% and 92.79%, respectively. Also, we obtained interesting findings concerning the quantity and quality of privacy relevant information published in the user reviews and their relation to the apps' behavior in reality indicating that user reviews are important and valuable source of information regarding the privacy behavior of mobile apps.

Keywords: smartphone apps, privacy, user review, mining threat, android

1. Introduction

By December 2018 the number of available apps in the Google Play Store was placed at 2.6 million [1], with more than 2 billion monthly active devices [2]. Accordingly and not surprisingly, there is a huge number of available user reviews. A mobile app user review is textual content that usually describes the properties of a mobile app (e.g. what is bad or good about it). These reviews are by default publicly available on app markets (e.g. Google Play Store, AppStore, etc.) and they are an important source of information for smartphone users when deciding to download a certain app. As they sometimes comprise privacy and security concerns, they can be also helpful for developers to gain knowledge about why users complain about privacy and security aspects of their apps. It has been shown that app developers need to be more knowledgeable about bad privacy practices that might happen due to the wrong definition and usage of Application Programming Interfaces (APIs) that are time to time revealed and claimed in app user reviews [3]. In addition, due to the complicated and unstructured nature of the written reviews, it is difficult and quite time consuming for developers to manually go through several thousands of reviews and to extract the desired privacy and security relevant information from them [4].

In this paper, we aim to tackle the aforementioned difficulties for mobile app developers. Therefore, we propose *Mobile App Reviews Summarization (MARS)* as a ML based system that exploits NLP and sentiment analysis techniques to assess the privacy behavior of apps by considering the privacy issues that have been reported by other users in the app-based reviews. Not only app developers, but also app users can benefit from our approach by understanding why other users complain about certain privacy and security aspects of apps. As a result, we design privacy indicators (interfaces) that reflect the existing knowledge about a specific app, so that users can better under-

☆This research work has received funding from the H2020 Marie Sklodowska-Curie EU project "Privacy&Us" under the grant agreement No 675730. This paper is published and copyrighted by *Elsevier Computers & Security Journal*.
Final published version available at: https://doi.org/10.1016/j.cose.2019.02.010

*Corresponding author
Email address: majid.hatamian.h@ieee.org (Majid Hatamian)

stand the privacy and security relevant information hidden in the reviews. As non-expert users may not have required knowledge to comprehend the information generated by *MARS*, we also propose an app behavior monitoring tool that can be jointly used with *MARS* to further increase the reliability of the whole analysis for normal users. Given the size and fast development of the Android market, we focus on the Google Play Store as one of the most valuable crowd-sources. The main challenge that *MARS* is going to address is to assess the privacy behavior of apps by determining how much privacy invasive information is observed in user reviews. Therefore, it ideally operates as a middle layer between app stores, mobile app developers and users as shown in Fig 1. App developers can benefit from such an approach by comprehending why users claim privacy deviated activities in their reviews that ultimately helps them to address the privacy issues of their apps.

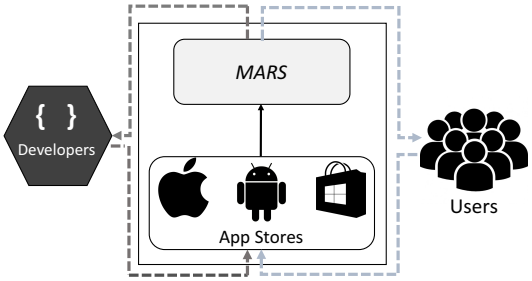


Figure 1: A high-level overview of *MARS*.

The investigation of the past research led to the following research questions:

RQ-1: How and why do mobile app users publish privacy relevant reviews?

RQ-2: How efficient and useful are ML, NLP and sentiment analysis techniques to extract privacy relevant information from app user reviews to help and support developers to understand the privacy relevant complaints about their apps in an efficient way?

RQ-3: Is it possible to detect potential privacy threats embedded in the app user reviews?

RQ-4: If the answer to **RQ-3** is positive, then how should we communicate the potential privacy risk to user to ease the process of informed privacy decision making?

RQ-5: How observable is the relation between app user reviews analysis and what the apps are actually doing in reality?

To provide answer to the aforementioned research questions, our work brings the following contributions:

C-1: Proposing a categorization of threats to the user's privacy and mapping these threats to the privacy related information extracted from user reviews to provide a better understanding for developers and end users;

C-2: Exploiting ML, NLP and sentiment analysis techniques to extract privacy relevant information from app user reviews, and accordingly, highlighting the usefulness of mining user reviews for analyzing the privacy and security aspects of apps that can be helpful for regulators, developers and users;

C-3: A novel summarization mechanism to ease the process of reading lots of lengthy reviews for users with interfaces that would alarm them about the potential threats that might arise due to the installation of certain apps;

C-4: Investigating the correlation and potential synergies between apps' user reviews analysis and the analysis of their behavior in reality by providing a novel app behavior monitoring system;

C-5: Examining the users' privacy perception and their reaction to *MARS* by conducting a user study.

The rest of this paper is organized as follows. Section 2 introduces existing relevant works in the literature. Section 3 describes how *MARS* is methodologically and practically designed and developed. Section 4 shows the results concerning the performance and functionality of our proposed approach. This section also provides several calls for actions and highlights further lines of research as the potential future work and details the limitations of our work. Finally, we present the main conclusions in Section 5.

2. Related Work

There is a diverse number of existing works concerning privacy and transparency enhancement of mobile apps. According to our specific application domain and to provide a clear structure, we categorize the existing works into two main categories: (1) App behavior analysis methods: Such methods lie on the fact that they cautiously look for privacy deviated behavior after app installation. As a result, they provide mechanisms to help users better understand what their installed apps are doing, e.g. which sensitive resources are being accessed and for which purpose; (2) User reviews mining methods: These methods are aimed at treating the smartphone users' privacy before app installation. Thus, through mining user reviews on app markets, they provide mechanisms and methodologies to enable users to compare apps in terms of privacy aspects and to support them for informed privacy decision making before app installation. The presented work in this paper belongs to this category. In what follows, we provide an overview of the current relevant state-of-the-art.

2.1. Behavior Analysis Methods

In [5], the authors introduced a method to make smartphone apps more privacy-friendly through automated testing, detecting and analyzing privacy violations. They suggested the use of an automated privacy-testing system to explore an app's functionality, logging relevant events at multiple levels of abstraction as the app executes, and using these logs to accurately

characterize app behavior. There are also approaches based on fine-grained control over permissions and majority voting recommendations [6, 7, 8]. These approaches enable users to turn on and off the access to sensitive data or functionality (e.g. SMS, camera, microphone, contacts) on an app-by-app basis to determine whether they feel comfortable granting it or not. In fact, in such solutions, a privacy control approach is provided to enable selectively granting, denying or confining access to specific permissions. Nevertheless, such solutions must be complemented with additional mechanisms that will first enable users to better understand the behavior of apps and the privacy implications. Following this direction, the authors in [9], proposed to identify permission hungry apps by considering the set of permissions declared by apps in app stores, and making a comparison of the commonly used permissions to make users aware of apps asking for rare or too many permissions.

Enck et al. [10] investigated the privacy of smartphone apps by monitoring a set of sensitive permissions, e.g. location, storage, contacts and phone number. In a sample of 311 of the most popular apps downloaded from the Google Play Store, they found five apps implementing dangerous functionalities, and therefore, should be installed with extreme caution. Followed by this study, the authors in [11] aimed at understanding of smartphone apps security by proposing a decompiler which recovers Android apps source code directly from its installation image. They analyzed 21 million lines of recovered code from 1,100 free apps using automated tests and manual inspection and it shows the use/misuse of personal/phone identifiers and deep penetration of advertising and analytics networks. TaintDroid [12] is a method in which the behavior of 30 popular Android apps is studied. The analysis showed that two-third of the apps show suspicious handling of sensitive data and that 15 of them reported users' location to remote advertising servers. Although these are important works and provide insights for privacy researchers, but they do not consider the importance of app meta data analysis such as user reviews, privacy policy, manifest declaration, etc. Habib et al. [13] proposed an automatic framework to assess the trustworthiness of mobile apps. Their framework is structured on app's reputation and state-of-the-art static analysis tools. They evaluated their framework on a data set of some selected apps from the Google Play Store that revealed their approach outperforms the existing methods. However, the researchers did not study the privacy-friendliness aspects of mobile apps. Furthermore, they did not investigate the importance of privacy and security analysis of user reviews. In [14], the authors studied the compliance of accessing permissions by installed apps with regard to the users' expectation. They modified the Android OS to log whenever an installed app accessed a permission-protected resource and then gave modified smartphones to 36 participants who used them as their primary phones for one week. Afterwards, they showed various instances over the past week where apps had accessed certain types of data and asked whether those instances were expected, and whether they would have wanted to deny access. The results showed that 80% of the participants would have preferred to prevent at least one permission request, and overall, they stated a desire to block over a third of all requests. This is an important work that revealed the discrepancy between users' expectation and actual app behavior. One of the most relevant insights from their work is the essential need of providing transparency for the users.

2.2. User Reviews Mining Methods

In this section, we provide an overview of the relevant related work in the area of mining smartphone apps user reviews.

The work presented by Fu et al. [15] proposes a three-level basis. The first level concentrates on individual user reviews and tries to investigate the impact of each word on user's actual sentiments. In the second level, they apply Latent Dirichlet Allocation (LDA) on the aggregated user reviews corresponding to a certain app to infer why users dislike the app. The last level explores the potential user preferences regarding app types and provides guidelines for developers. Iacob et al. [16] studied the problem of automatic retrieval of mobile app feature requests from their user reviews. As the first step, they examined the extent to which the user reviews contain feature requests. They further used LDA to identify common topics across the feature requests. Their analysis on a data set of 136,998 online reviews showed that users' requests are mostly related to support for apps, frequent updates, new levels for game apps and more customization options. In [17], the authors studied the problem of automatic topic extraction from app user reviews. They used topic modeling techniques and assumed that the importance of a topic is proportional to the number of reviews it receives. By doing a case study on three popular Android apps, they extracted the most frequent topics for each app and they further validated these results by manual observations to show how their approach can diminish the manual effort of user reviews analysis. Oh et al. [18] presented an algorithm to detect informative reviews reflecting user involvements. They trained a Support Vector Machine (SVM) classifier on a data set of user reviews extracted from the Google Play Store to automatically classify them into functional bugs, functional demands and non-functional requests. Pagano et al. [19] performed an exploratory study over 1,126,453 reviews from 1,100 iOS apps. The main goal was to use statistical analysis and topic modeling to investigate how and when users publish reviews, with special focus on the review's content. Their experiments showed that the user reviews were mostly published shortly after new releases. Regarding the popular topics, they found user experience, bug reports and feature requests as the most frequent topics recurring in the reviews.

Guzman et al. [20] presented a system to ease the analysis of app user reviews for developers. They first employed NLP techniques to extract the most relevant app features together with the user opinions about them. The results confirmed the positive influence of sentiment analysis on the overall extraction of the most frequently mentioned features. In [21], a method has been proposed to investigate the most informative user reviews from their large and rapidly increasing pool to help app developers to improve their apps. The authors used a review ranking scheme to prioritize the informative user reviews. Furthermore, a filtering process is utilized to filter out non-informative reviews.

In [22] and [23] the authors studied the problem of privacy invasive apps, treating the classification of user reviews on the Google Play Store as a supervised learning task. Their goal was to detect different types of privacy and security relevant information based on the collected data comprised of user reviews. Their experiment showed that their classification methods provide performance as high as 72,63% and 81,26%, respectively.

Panichella et al. [24] used text mining and sentiment analysis for the purpose of software maintenance and evolution to classify app user reviews into different categories. The results of their study enabled them to identify and extract useful information that would be applicable for app developers. In [25] and [26], the authors were motivated by what smartphone users do complain about. Therefore, based on ML techniques they examined 20 Android and iOS apps and their respective user reviews to reveal the most common complaints. The results of their study indicated that users talk about diverse topics in the reviews, including compatibility issues, crashing, feature removal requests, network problems, etc.

Ciurumelea et al. [27] introduced an approach to assist app developers to analyze the user reviews. They first manually labeled 1,566 user reviews related to distinct taxonomies, including compatibility, usage resources, pricing, protection and complaint. Next, they employed the Gradient Boosted Regression Trees (GBRT) model to classify each user review into one of these categories. They validated the performance of their approach by conducting an empirical study on the reviews of 39 apps. In [28], the authors used static code analysis to extract permission features mentioned in the app reviews. The goal was to study the relation between app user reviews and security developments in apps. They found that almost half of all privacy oriented changes in apps are related to third-party library code. It was also revealed that apps adopting run-time permissions receive a higher number of security relevant user reviews that would be helpful for developers to revise their implementation strategies.

These are interesting works and they provide insights for mining user reviews. Nevertheless, there are still few works focusing solely on the privacy issues identified in app user reviews. Also, neither of these works target both developers and users at the same time. As the presented work in this paper belongs to this category of relevant works, we found it as of particular importance to provide a detailed comparison with the reviewed works as shown in Table 1.

3. Mobile App Reviews Summarization (MARS)

In this section, we introduce *Mobile App Reviews Summarization (MARS)* as a new approach for the summarization of app user reviews on app markets. In what follows, we elaborate the methodologies that we followed for data collection and corpus preparation (Section 3.1) and the automated user reviews analysis that benefits from ML techniques (Section 3.2). Fig. 2 summarizes a high level architecture of the proposed components for *MARS*.

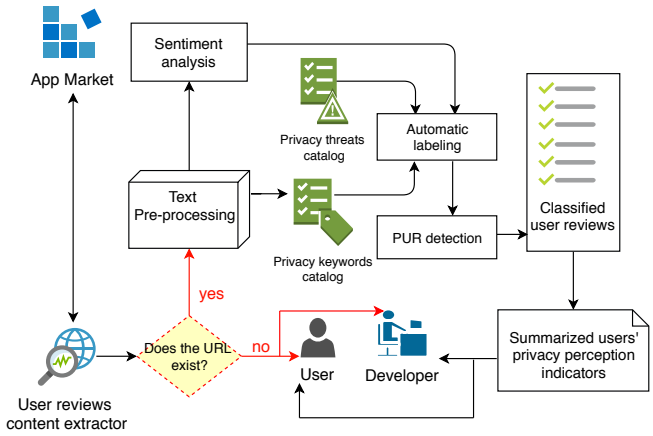


Figure 2: A high level architecture of *MARS*.

3.1. Data Collection and Corpus Preparation

This subsection describes the steps and logic that we considered for data collection (Section 3.1.1), the extraction of the privacy keyword catalog (Section 3.1.2), pre-processing done on the collected data set (Section 3.1.3), and it lastly shows how sentiment analysis is leveraged for the sake of negative and positive privacy relevant statements identification (Section 3.1.4).

3.1.1. Data Collection

To make our scope as narrow as possible, we first analyzed a large number of apps, which enabled us to identify the app categories that mostly target privacy sensitive permissions and further analyzed the top 20 apps per identified category (based on the search results on the Google Play Store). To this end, we used the scraper in [29] to crawl the Google Play Store. We first collected information from 981,075 apps, and for each app we retrieved the app URL, app category and the set of permissions (as defined in its `AndroidManifest.xml`). As a result of this analysis, 142 distinct types of permissions were extracted. By analyzing the lists of privacy sensitive permissions [1] that each app requests, we focused on the app categories (top 10 out of 42 categories, as shown in Fig. 3) that have the maximum number of apps that request at least two privacy sensitive permissions (such as `READ_CONTACTS`, `CAMERA`, etc.) and Internet connection access.

As a second step, we selected the top 20 apps (based on the search results on the Google Play Store) per app category (in total, 20 apps $\times$ 10 categories $=$ 200 apps). We used the list of URLs of the 200 apps as an input in order to extract their associated user reviews (max 4,500 reviews per app). We used [29] to collect these data during a one-month period of time (September 2018). Our initial data set was then comprised of 812,899 users reviews associated to the 200 apps within 10 app categories.

3.1.2. Privacy Keyword Catalogue Extraction

The aim of this process was twofold: (1) to come up with a more fine-grained data set *"tuned data set"* that included

[1] https://developer.android.com/guide/topics/permissions/overview

Table 1: List of reviewed works with their respective properties (P: Precision, R: Recall, SA: Statistical Analysis, MA: Manual Analysis).

No.	Author(s)	Method	Privacy Focus	Threat Finding	Sentiment Analysis	Performance	# Samples
1	Fu et al. (2013)	LDA	No	No	Yes	N/A	13,286,706
2	Iacob et al. (2013)	LDA	No	No	No	P(85%), R(87%)	136,998
3	Galvis Carreño et al. (2013)	K-Median, ASUM	No	No	Yes	P(75%), R(19%), F-score(30.32%)	2,651
4	Oh et al. (2013)	SVM	No	No	Yes	P(89%), R(81%), F-score(85%)	1,711,556
5	Pagano et al. (2013)	SA	No	No	No	N/A	1,126,453
6	Guzman et al. (2014)	LDA	No	No	Yes	P(58%), R(52%), F-score(54%)	32,210
7	Chen et al. (2014)	ASU, LDA	No	No	No	Hit-rate (70%) and NDCG@10 (50%)	181,097
8	Cen et al. (2014)	ILR	Yes	Yes	No	F-score(72.63%)	36,464
9	Kong et al. (2015)	Sparse SVM	Yes	Yes	No	P(80%), R(82%), F-score(81%)	19,143
10	Panichella et al. (2015)	Bayes, SVM, LR, J48, ADTree	No	No	Yes	P(75%), R(74%)	32,201
11	Khalid et al. (2015)	MA	No	No	No	N/A	6,390
12	Mcilroy et al. (2016)	SVM, J48, Naive Bayes	No	No	Yes	P(50%), R(62%), F-score(55%)	N/A
13	Ciurumelea et al. (2017)	GBRT	No	No	No	P(51%), R(79%)	1,566
14	Nguyen et al. (2018)	SVM	Yes	No	No	AUC(93%)	5,527
15	**Proposed Approach**	**SVM, LR, KNN, DTree, ETree**	**Yes**	**Yes**	**Yes**	**F-score(93%)**	**812,899**

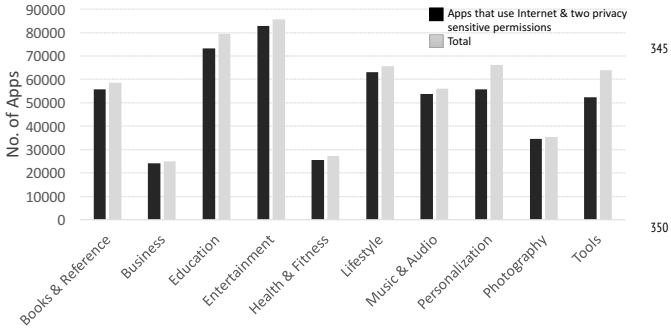


Figure 3: Top 10 app categories in terms of privacy sensitive permission requests.

privacy relevant claims; (2) to identify relevant keywords that would ease and smooth the process of automatic labeling and classification (see Section 3.2). To create the tuned data set, we first chose "privacy" and "security" as the two most relevant keywords and filtered the initial data set. We then employed `FreqDist` class in NLTK [30] to infer the frequency of keywords. We followed an iterative approach where the steps depicted in Fig. 4 were repeated on the original data set. At each iteration we used the extracted privacy keyword catalog in the filtering phase (in addition to the initial two keywords), see Fig. 4(b).

3.1.3. Pre-processing

To increase the quality of our data set to those user reviews including privacy relevant claims, the filtering and preprocessing of data was critical, especially due to the large amount of the collected user reviews consisting of non privacy related information (i.e., reviews about functionality, etc.). We exploited standard techniques of NLP for filtering and stemming, thus, we used NLTK [30] based on Python, and performed the following tasks:

1. Tokenization: Each user review is split into several tokens to later ease the process of stemming and removing stop words;
2. Removing stop words: To boost the algorithm processing time, we removed stop words (e.g., "the", "on", "is", "at", etc.);
3. Stemming: We applied stemming on all user reviews in order to reduce the number of words and to improve the results of the NLP processes. Stemming is a common NLP technique to identify the word's root, and it is essential to make words such as "argue", "argued", and "arguing" all match to the single common root "argu".

Both processes (pre-processing of data and keyword catalog extraction) resulted in a higher quality (reduced) data set aimed at increasing the efficiency of both, the automatic user reviews labeling (see Section 3.2.2) and the supervised learning tasks (see Section 3.2.3).

3.1.4. Discrimination of Positive and Negative Privacy Statements

It is generally assumed that user reviews containing privacy/security relevant statements may comprise negative sentiment. However, after manual observations, we also identified positive claims, thus, in order to avoid an unfair evaluation of apps, we decided to apply sentiment analysis techniques to better discriminate the privacy relevant reviews. Furthermore, this step is done to possibly avoid potential collusion attacks where a rogue developer convinces, incentivizes, or tricks users into leaving positive or negative reviews for an app. Accordingly,

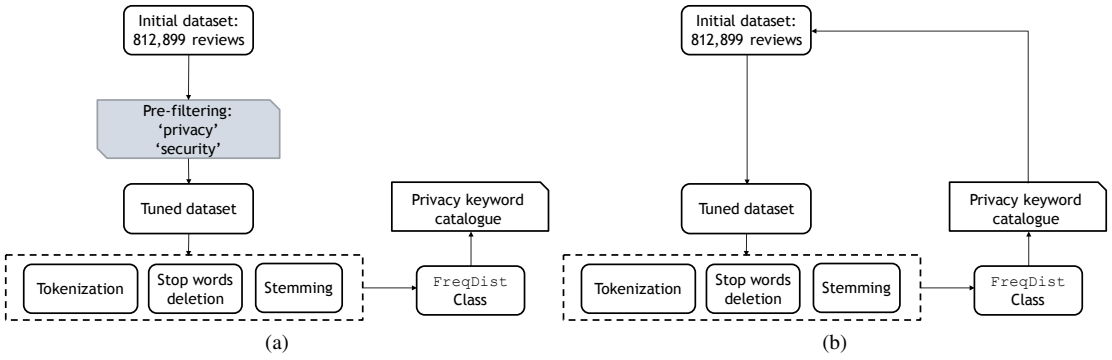


Figure 4: Privacy keyword catalog extraction procedure.

we used *Valence Aware Dictionary and sEntiment Reasoner (VADER)* [31] - an open source lexicon and rule-based sentiment analysis tool attuned to sentiments expressed in social media. It detects the polarity (positive, neutral, negative) and the sentiment intensity with a good performance and it does not require any training data [31]. VADER relies on a dictionary which maps lexical features to emotion intensities called sentiment scores, which are obtained by summing up the intensity of each word in the text. In our approach, for each user review, the sentiment score corresponding to each of its words is calculated on a scale of $[-4, +4]$, where -4 is the most negative and $+4$ is the most positive. The global sentiment score per user review is then the sum of the sentiment score of each of its words. We then normalize the total sentiment score by $x/\sqrt{x^2+\alpha}$ to map and represent the score into a value between $[-1, +1]$ using an alpha that approximates the max expected value, where x is the sum of the sentiment scores of the user review's words and α is the normalization parameter.

3.2. Automated User Reviews Analysis

In this subsection, we elaborate the preliminaries and requirements that are essential for *MARS* to operate (Sections 3.2.1 and 3.2.2) and elucidate how ML techniques help *MARS* to function (Section 3.2.3).

3.2.1. Privacy Threat Identification

We detect not only a privacy relevant user review, but also the privacy threat hidden in it. To this end, we performed a literature research [32, 33, 34, 35, 36] in order to identify the most relevant privacy threats in the context of smartphone ecosystems as shown in Table 2. These threats are further used as the input for the automatic user reviews labeling (Section 3.2.2) and supervised classification algorithm (Section 3.2.3).

3.2.2. Automatic User Reviews Labeling

The classification of user reviews is a supervised learning task that needs the provision of labeled training and testing data. In this work, we propose an automatic solution that avoids the need of manual labeling. To this end, we use GloVe [37] model, which represents words that have the same meaning in the form of vectors. The algorithm first collects word co-occurrence statistics in the form of word co-occurrence matrix X. Each component X_{ij} of such matrix represents how often word i appears in context of word j. We represent $\sum_k X_{ik}$ as the number of times any word appears in the context of word i. As a result, the probability that word j appears in the context of word i is defined by $P_{ij} = P(j|i) = X_{ij}/X_i$. It then goes through the data set in the following manner: for each term it looks for context terms within some area defined by a *window_size* before and after the term. It is important to note that we give less weight for more distant words. It then determines $w_i^T w_j + b_i + b_j = \log(X_{ij})$ as the soft constraints for each word pair, where b_i and b_j are scalar biases for the main and context words, and w_i and w_j represent the vectors for them, consecutively. Accordingly, the following weighted least squares regression model which operates as the cost function is determined to be minimized:

$$J = \sum_{i=1}^{V} \sum_{j=1}^{V} f(X_{ij})(w_i^T w_j + b_i + b_j - \log X_{ij})^2, \qquad (1)$$

where V is the size of the vocabulary, and $f(X_{ij})$ is a weighting function which prevents learning only from excessively common word pairs defined as follows:

$$f(X_{ij}) = \begin{cases} (X_{ij}/x_{max})^\alpha & \text{if} \quad X_{ij} < x_{max} \\ 1 & \text{otherwise,} \end{cases} \qquad (2)$$

where x_{max} and α are by default set to 100 and 3/4, respectively.

Word Vectors for Privacy Threats and User Reviews. We give as the input the words in our data set to the GloVe and it returns their word representation (in the form of vector of 300 components). As the next step, we need to construct the vector of words for each privacy threat (which is of size 300 as well) to represent that privacy threat. For this purpose, we used the tokens from our privacy keyword catalog (see Section 3.1.2). In order to construct the vector of words for each privacy threat, we get from the GloVe a vector X_1 for $Token_1$, vector X_2 for $Token_2$, ..., and vector X_n for $Token_n$. For each $Token_i$ $(1 \leq i \leq n)$ we take the average vector by $X = (\sum_{i=1}^{n} X_i)/n$, where X is a 300 dimensional vector and it holds the contextual meaning of all n

Table 2: Identified threats.

#	Threat	Description
T1	Tracking & Spyware	Allows an attacker to access or infer personal data to use it for marketing purposes, such as profiling or targeted ads. It covers untargeted collection of personal information as opposed to targeted surveillance.
T2	Phishing	An attacker collects user credentials (such as passwords and credit card numbers) by means of fake apps or (SMS, email) messages that seem genuine. Smartphones have a smaller screen, which means that attackers can more easily disguise trust cues that users rely on to decide on submitting credentials; e.g. cues that show whether the website uses SSL. Also they provide additional channels that can be used for phishing, e.g. SMS. Users may be less cautious about SMS phishing messages, e.g. clicking on unwanted links, etc. that leads to privacy and security loss.
T3	Unauthorized Charges	The hidden and unauthorized charges through registration to a premium service AND/OR installation a certain app that lead to privacy loss and monetary consequences.
T4	Unintended Data Disclosure	Users are not always aware of all the functionality of smartphone apps. Even if they have given explicit consent, users may be unaware that an app collects and publishes personal data. Location data, for example, is often used in social networks - in messages or uploaded photo metadata, in augmented reality apps, micro-blogging posts, etc. Most apps have privacy settings for controlling how and when location data is transmitted, but many users are unaware (or do not recall) that the data is being transmitted. Malwares are also categorized in this category as they target mobile phones by causing the collapse of the system and loss or leakage of confidential information.
T5	Targeted Ads	Refers to unwanted ads and push notifications.
T6	Spam	Threat of receiving unsolicited, undesired or illegal messages. Spam is considered an invasion of privacy. As we would not want a stranger knocking on our door, calling us on the phone, or following us down the street, receiving unsolicited messages/emails is an infringement of the right to be left alone. The receipt of spam can also be considered a violation of our right to determine for ourselves when, how, and to what extent information about us is used.
T7	General	Comprises all the issues that are not categorized into other threats, such as permission hungry apps, general privacy and security concerns, etc.

tokens. For instance, if we take the token "*password*" and the token "*hack*", then we add vector of "*password*" to the vector of "*hack*" and we get as a result a vector that is very similar to the vector of token *theft* (T2).

Privacy Text Expansion. The aim of this step was to find as many privacy and security tokens for each privacy threat as possible. Thus, for each privacy threat we searched in the GloVe vocabulary for the N most similar words. This process is known as text expansion, meaning that we have only few words (described in Section 3.1.2) but we aim to expand them to learn other analogous words. To perform this, the algorithm takes cosine similarity between the vector of each privacy threat and the vector of a particular word from the vocabulary, and then it gives us N words that have the highest value of cosine similarity. For example, given $glove.find\_n\_most\_similar(text =$ "privacy", $n = 3$), the output is "*surveillance=0.8*", "*permission=0.7*", and "*functionality=0.4*".

Final Data Set. As a result of the aforementioned steps, we automatically labeled 2,896 privacy relevant user reviews. In order to validate the accuracy of the automatic labeling procedure and to avoid potential errors (false positives/negatives) that might happen due to the automatic nature of this approach, we involved three privacy experts who performed a manual validation of the automatic labeling. In other words, these experts were assigned to check/validate whether the automatic labeling has been correctly done (how accurate it was). Accordingly, they manually checked the labels assigned to each user review. Finally, 2,412 (out of 2,896, with an accuracy as high as 83.28%) privacy relevant user reviews were confirmed and used as the input for the classifier.

3.2.3. Privacy-relevant User Reviews (PUR) Detection

This section describes the methodology used to classify the user reviews, namely, feature extraction and classification algorithm.

Feature Extraction. We use *term frequency-inverse document frequency (tf-idf)* feature [38] to represent each user review for the PUR detection. Tf-idf looks at a normalized count where each word count is divided by the number of documents this word appears in. Let C be the set of m user reviews $C = \{c_1, \ldots, c_m\}$ consisting of the training and testing sets t_1 and t_2 including their respective privacy threat classes. With tf-idf as the feature and annotation with six labels (privacy threats), the data set can be represented by the set of tf-idf feature vectors and the set of label vectors for each c_i ($1 \leq i \leq m$) as $f = \{f_{c_1}, ..., f_{c_m}\}$ and $l = \{l_1, ..., l_j\}$, respectively. Given a collection of user reviews C, a word w, and an individual user review $c_i \in C$, we calculate

$$w_{c_i} = f_{w,c_i} \times \log(|C|/f_{w,C}), \tag{3}$$

where f_{w,c_i} equals the number of times w appears in c_i, $|C|$ is the size of the corpus, and $f_{w,C}$ equals the number of times w appears in C. The higher the w_{c_i} value, the more discrimination power is achieved, meaning that user reviews with high w_{c_i} indicate that w is an important word in c_i.

Classification Algorithm. In order to implement the supervised classification algorithms for the PUR detection, we exploited

Table 3: Performance measures of the classification algorithm.

Classes	Recall	Precision	F-score
Tracking & Spyware	0.7766	0.8846	0.8214
Phishing	0.8487	0.8813	0.8695
Unauthorized Charges	0.7806	1.00	0.7400
Unintended Data Disclosure	0.7301	0.9067	0.7001
Targeted Ads	0.9971	0.9374	0.9663
Spam	0.8132	0.7514	0.8074
General	0.8548	0.8750	0.8696
Overall	0.9130	0.9484	0.9279

the scikit-learn open source library for Python [39]. We employed various algorithms, namely, *Decision Trees, Extra Trees, K-Nearest Neighbours (KNN), Random Forest, Logistic Regression (LR),* and *Support Vector Machine (SVM)* classifiers [40] in order to select the one with best performance. Our experiments revealed that LR has the highest accuracy among others, mainly because LR is intrinsically simple, less prone to over-fitting and it has low variance [41].

4. Experiments and Results

4.1. Classification Performance Results

We used $CountVectorizer$ and $TfidfTransformer$ packages in scikit-learn for the feature extraction phase. We then split the data set into training and testing data (70% for training and 30% for testing). Using scikit-learn we exploited several classification algorithms (see Section 3.2.3). We observed LR outperforms others, therefore, we only show the results for LR. We used recall, precision and F-score metrics to evaluate the performance of the classifier. The values of these metrics show how well the *MARS*'s results correspond to the annotated results. If we show the values of true positives, false positives, true negatives, and false negatives by TP , FP , TN and FN, then we have:

$$\text{Recall} = \frac{TP}{TP + FN}. \tag{4}$$

$$\text{Precision} = \frac{TP}{TP + FP}. \tag{5}$$

$$\text{F} - \text{Score} = 2 \cdot \frac{\text{Recall} \cdot \text{Precision}}{\text{Recall} + \text{Precision}}, \tag{6}$$

where TP , FP , TN and FN show the correctly identified, incorrectly identified, correctly rejected and incorrectly rejected user reviews, respectively. Table 3 shows the values for the aforementioned metrics corresponding to each privacy threat. The observation is that the overall recall and precision values are of 91.30% and of 94.84%, respectively. Moreover, the values obtained for F-score show the good performance of our approach.

In Fig. 5 we show the results regarding the computational performance of *MARS* in terms of time as applied to a new set of user reviews for top 20 apps (at the time of writing this paper) in *Entertainment* app category. We chose this category as it

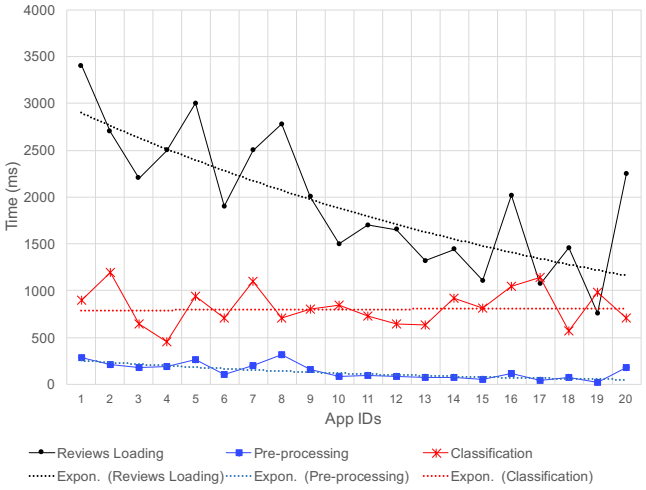


Figure 5: Computational performance of *MARS* (in terms of time) per step.

comprises the highest number of apps requesting sensitive permissions (see Section 3.1.1 and Fig. 3). We observed that the review loading step takes more time than others which highly depends on different factors. First, the `node.js` script that is used to retrieve the user reviews has to integrate all the collected reviews for a certain app, meaning that it is not only about data collection, but also data integration. More importantly, the Google Play Store only shows 40 user reviews per page. As a result, it imposes more time overhead to retrieve and achieve the maximum available number of reviews in each page. Additionally, the review loading step is extremely dependent on the Google Play's response time and the length of each retrieved review. All in all, *MARS* only requires negligible time to process the user reviews and to extract privacy relevant information which is notably less than manual task of reading reviews that can support developers and users to save a considerable amount of time required to read and understand fuzzy and lengthy reviews.

4.2. Comparative Analysis

In this subsection, we compare our approach with the most relevant existing works in the literature, namely *CDCE* [22] and *AUTOREB* [23]. To start with this comparative analysis, we first show the results of performance measurement. Table 4 shows the performance of our approach against the aforementioned approaches in terms of precision, recall and F-score. It is observed that *MARS* substantially outperforms both approaches (unfortunately the values of recall and precision were not available for CDCE method). This can be interpreted in different ways. Firstly, dissimilar to CDCE and AUTOREB, we used an automatic labeling system together with a manual validation that could eliminate the potential errors happening in traditional manual labeling tasks (e.g. Amazon Mechanical Turk) and would increase the accuracy. Secondly, we used sentiment analysis as an important step that helped us to increase the granularity of our data set for the privacy relevant user reviews detection. To the best of our knowledge, this is the first time that

Table 4: Comparative analysis of CDCE, AUTOREB and *MARS*.

Metric	CDCE	AUTOREB	**SUPPS**
Precision	N/A	0.8010	**0.9484**
Recall	N/A	0.8246	**0.9130**
F-score	0.7263	0.8126	**0.9279**

sentiment analysis is employed for the sake of privacy analysis of user reviews in the context of smartphone apps.

4.3. MARS Results

4.3.1. Sample Classified User Reviews by MARS

To gain a better understanding of the classified user reviews, Table 5 shows some examples regarding the strength of *MARS* in distinguishing different types of user reviews together with their corresponding privacy threat (shown by T).

Table 5: An example of classified user reviews.

#	Sample user review	T
1	*You don't need to spy on my activities outside of this app. they don't care about their customers, they want to ruin the device with horrible bloatware spyware*	T1
2	*Im still getting warnings that my phone is infected with virus after i update and scan again. If its not going to work why download it. I have very limited memory to use. No need to download stupid apps that dont work*	T2
3	*Cheating Y the hell.. u cut my 50 rupees for nothing.. i just enter my card details and u cut my money without asking me.. i want it back*	T3
4	*SHit!Takes control of device.. why my photo is there???!!*	T4
5	*Ads are terrible Sorry but the ads are comparing to the website really irritating.*	T5
6	*Simple interface to use with plenty of features - but pop ups*	T6
7	*Dangerous! requires unnecessary access to sensitive permissions! Uninstalled*	T7

4.3.2. Distribution of Privacy-relevant User Reviews

A detailed view of the distribution of privacy-related reviews considering their associated privacy threat and category is shown in Fig. 6. The most common reported threats are *Spam & Ads (T5)*, *General Privacy Issues (T6)*, and *Unauthorized Charges via Premium Services (T3)*, however, *Profiling & Tracking (T1)* is the least reported privacy concern within all the app categories. Not surprisingly, app category *Lifestyle* is the most reported (which comprises a huge number of social network apps). One important finding is that users worryingly report privacy concerns regarding apps within categories that mostly target users' general interests such as *Entertainment*, *Tools*, *Music & Audio*, etc. This is interesting, as it is unexpected for such apps to aggressively invade users' privacy. This reveals that app categories (like *Tools*) that are supposed to not interactively access privacy sensitive information, are still accessing such resources which in turn has raised the users' concern by posting privacy related reviews.

In Fig. 7, we depicted the frequency of privacy-related reviews per app category. The results show that *Lifestyle*, *Entertainment* and *Tools* categories have the highest number of privacy-related reviews, 528, 318 and 264 reviews published by the users, respectively. Moreover, *Education*, *Book & References* and *Photography* categories comprise the minimum number of privacy-related reviews with a portion of 79, 86 and 154 reviews.

4.3.3. The Most Repetitive Permissions

In overall, we found 3,201 statements corresponding to ten sensitive permissions while some of the PURs comprise multiple statements referring to a certain permission. Fig. 8 shows the top 10 most stated permissions within PURs. The bar chart depicts that the most repetitive permissions are storage (e.g. complaining about access to photos, videos, etc.), location and phone state (e.g. complaining about access to outgoing calls, phone numbers, etc.) being mentioned 610, 533 and 497 times, respectively. In contrast, calendar, sms access and microphone permissions are the least repetitive permissions. It is interesting to see a diverse number of complaints concerning the sensitive permission requests by health-based apps

4.4. Privacy Perception Vs. Reality: Case Study on Health & Fitness Category

As *MARS* enables us to analyze the users' perception considering the published user reviews, we were interested to investigate whether there is any positive/negative correlation between user reviews (perception) analysis and what the apps are doing in reality.

4.4.1. App User Reviews Analysis for the Health & Fitness Category

By using *MARS*, we discovered that some app categories (e.g. *Health & Fitness*) that do not need to excessively request or access privacy sensitive information, are getting more privacy related user reviews, meaning that the users are publishing a considerable amount of privacy related claims regarding such app categories. In Table 6 we report the identified privacy threats associated with the apps in category *Health & Fitness* (shown by ✓) which gives us a more comprehensive view on the details of threats corresponding to each app detected by *MARS*. Although there are other app categories containing a higher number of privacy relevant reviews (e.g. *Lifestyle*), for our analysis we decided to focus on *Health & Fitness* category. Such selection is rationalized as follows: (1) Researchers have raised serious privacy and security concerns resulted from using invasive health-based apps [42, 43, 44, 45]; (2) Health-based apps are sometimes underestimated by the users. As compared with other popular app categories such as *Lifestyle*, users are not well-aware of the potential negative consequences of using privacy invasive health-based apps. For instance, in the early 2018, already people were informed about Facebook-Cambridge Analytica data privacy scandal [46]. Hence, it is generally believed that lifestyle-based and social networking apps are the only main potential sources of privacy violations; (3) As a result of extreme proliferation of gadgets and physical activity trackers (such as FitBit), users are currently surrounded by such technologies. Such technological trend is highly dependent on wireless communications between gadgets and

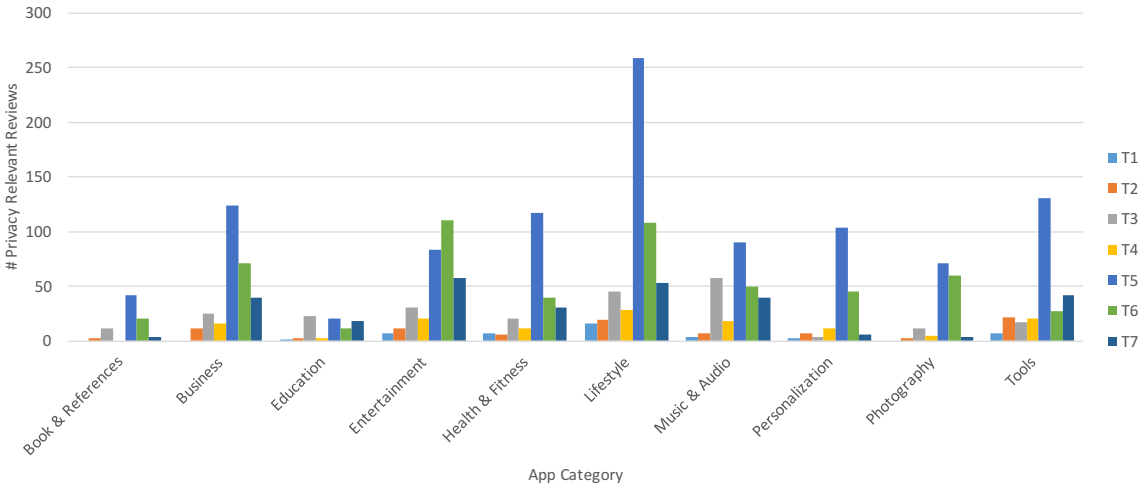


Figure 6: Distribution of privacy relevant reviews by category and privacy threat.

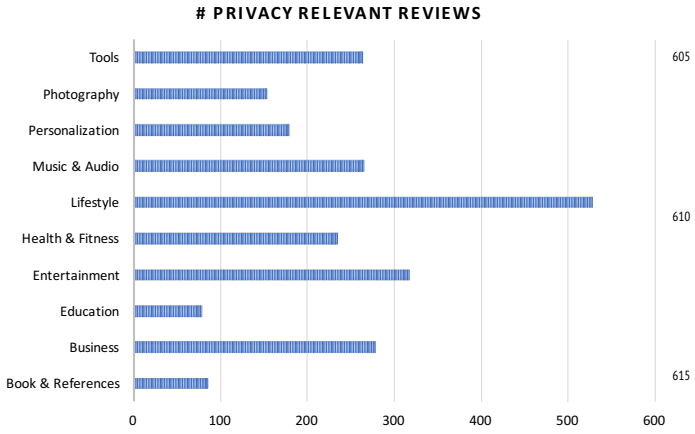


Figure 7: Distribution of privacy relevant user reviews per app category.

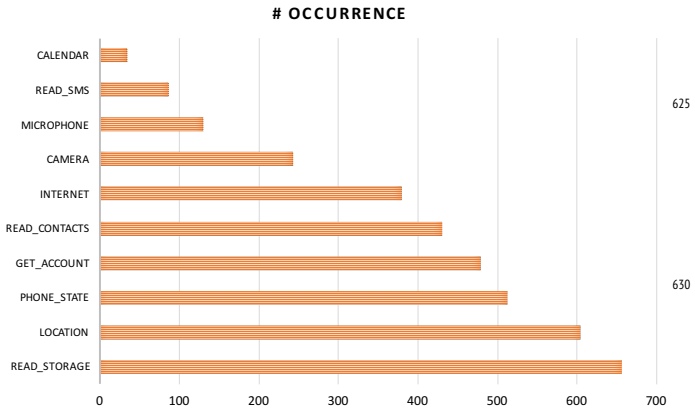


Figure 8: Top 10 most stated permissions in user reviews.

smartphones (i.e. health/fitness-based apps) that may potentially impose privacy risks; (4) In contrast to other general purpose app categories, health/fitness-based apps are directly dealing with special users' sensitive data such as body sensors which are classified as highly sensitive data (Art. 4(13), (14), (15) and Art. 9 and Recitals (51) to (56) of the General Data Protection Regulation) [35].

This backs up the point that users must pay careful attention before downloading such apps and further shows the applicability and usefulness of *MARS* in discrimination of such app categories. Regardless of the choice of the research area, *MARS* can be adapted and applied to other smartphone ecosystems (e.g. App Store). Also, as it is trained with a significant number of samples, it can be simply and widely used as a standalone (without the need to be trained again) to analyze the privacy aspects of mobile apps considering their user reviews.

4.4.2. *What Do Apps Do in Reality: Behavior and Network Output Analyses for the Health & Fitness Category*

As the obtained results regarding the analysis of the user reviews for the *Health & Fitness* category were to a certain extent interesting, we decided to perform a case study to have a more in-depth analysis regarding the validity and reliability of our results. Thus, we were interested to analyze the privacy behavior of these apps in reality plus the network output traffic.

App Real Behavior Analysis. We designed and implemented a behavior monitoring tool for Android apps [47, 48]. When it comes to normal users, we recommend to use such behavior monitoring tool jointly with *MARS* to increase its reliability. The tool benefits from two main components:

1. `LogReader`: The `LogReader` component is responsible to read the AppOps Manager [49] within a certain time interval including all the resource accesses done by the installed apps (e.g. access to sensitive resources like `CAMERA`, `READ_CONTACTS`, `LOCATION`, etc.). To collect the logs, a timer is sent to the `PermissionUsageLogger` service periodically. When it is received, the logger queries

Table 6: List of apps in category *Health & Fitness* with their respective identified privacy threats.

No.	App url	Tracking & Spyware	Phishing	Unauthorized Charges	Unintended Data Disclosure	Targeted Ads	Spam	General
1	com.sec.android.app.shealth	✓	×	×	×	✓	✓	×
2	com.google.android.apps.fitness	×	×	×	×	✓	✓	×
3	com.sillens.shapeupclub	×	×	×	×	×	×	✓
4	cc.pacer.androidapp	×	✓	×	×	✓	×	×
5	com.myfitnesspal.android	×	×	×	×	✓	×	×
6	pedometer.steptracker.calorieburner.step	×	×	×	×	✓	✓	×
7	com.stt.android	×	✓	✓	✓	✓	✓	×
8	com.fitnesskeeper.runkeeper.pro	✓	✓	✓	✓	✓	✓	✓
9	com.fitbit.FitbitMobile	✓	×	✓	✓	✓	✓	✓
10	com.nike.plusgps	×	×	×	×	✓	×	✓
11	com.runtastic.android	✓	×	✓	✓	✓	×	✓
12	com.popularapp.sevenmins	×	×	×	×	✓	✓	×
13	com.popularapp.thirtydayfitnesschallenge	×	×	×	×	✓	✓	×
14	si.modula.android.instantheartrate	×	✓	✓	✓	✓	×	✓
15	com.playsimple.fitnessapp	×	×	×	×	✓	✓	×
16	com.mapmyrun.android2	×	×	✓	×	×	✓	×
17	com.macropinch.hydra.android	✓	✓	×	✓	✓	×	✓
18	com.fitness22.running	×	×	×	✓	✓	×	✓
19	com.endomondo.android	×	×	✓	×	✓	✓	×
20	comm.cchong.BloodAssistant	✓	✓	✓	×	✓	×	✓

the AppOps service that is already running on the phone for a list of apps that have used any of the operations we are interested in tracking. We then check through that list and for any app that has used an operation more recently than we have checked, we store the time at which that operation was used in our own internal log. These timestamps can then be counted to get a usage count. The data produced by this component is then processed and fed into the `DataMining` component.

2. `DataMining`: The `DataMining` component is supposed to behaviorally analyze the installed apps by getting help from the results obtained from the `LogReader` component. This is done according to a rule-based approach which is supposed to increase the functionality and flexibility of our `DataMining` component. Consequently, we have defined a set of privacy deviated behavior detection rules (based on two rounds of privacy expert discussions) that are aimed to analyze the privacy behavior of the users' installed apps. We initially defined a set of sensitive permissions (introduced by Android[2]) and we mainly analyze the accesses to these resources. For example, imagine that a device's screen is off and it is in horizontal orientation (and user does not talk on the phone, meaning that the `AUDIO` permission is not being used). In such situation, we assume that user does not use the phone (e.g. the phone lies on the desk) and if one of the sensitive resources is accessed by a given installed app, we record this and report to the user about the detail of resource access (date, time and reason together with a short explanation). Therefore, the users can transparently manage their resource accesses.

Under certain circumstances, access to a certain personal resource cannot be justified. For example, accessing `CAMERA` under normal circumstances is perfectly legitimate, but if the phone was lain on the table, accessing the camera needs to be more investigated. Thus, while implementing the monitoring tool, we paid special attention to the following elements to discern which resource access might be meaningful (needed) :

- Device's Orientation: This gives us information about where the device is located, e.g., if the screen is down or up. We register a `Listener` who is listening to changes in the accelerometer that ultimately gives us the values for x, y and z.
- Screen State: The screen state describes whether the device's screen is on or off at a certain time. As long as a scan is running, we register a `Receiver` for the events `ACTION_SCREEN_ON` and `ACTION_SCREEN_OFF`.
- Proximity Sensor: The screen state alone, however, is not meaningful enough, as it may happen that the screen is

[2]https://developer.android.com/guide/topics/permissions/requesting.html

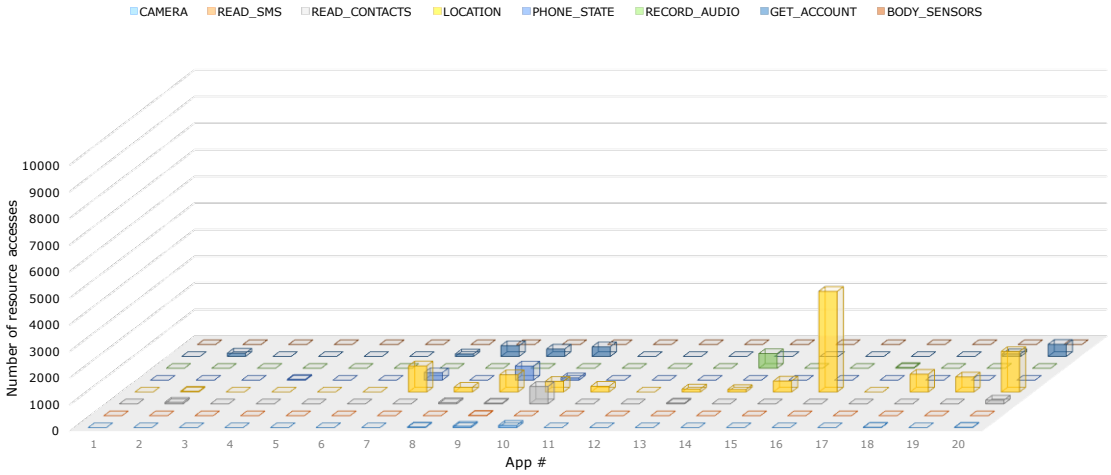
indeed off but certain personal resources may still be accessed (e.g., when talking on the phone, the screen turns off when the phone is approaching the ear, but access to RECORD_AUDIO is justified at this time). Therefore, we read the proximity sensor to indicate whether an object is within a defined range of the mobile phone. We access it through SensorManager and the associated listener is called as soon as an object enters or leaves the defined range (the range varies from device to device, but on average, it is around 5cm).

- App State: We also consider the app state (at the time of access to a certain resource) as an important element while monitoring the apps' behavior. We distinguish the following app states: SYSTEM_APP, PRE_INSTALLED_APP, INACTIVE, BACKGROUND and FOREGROUND.

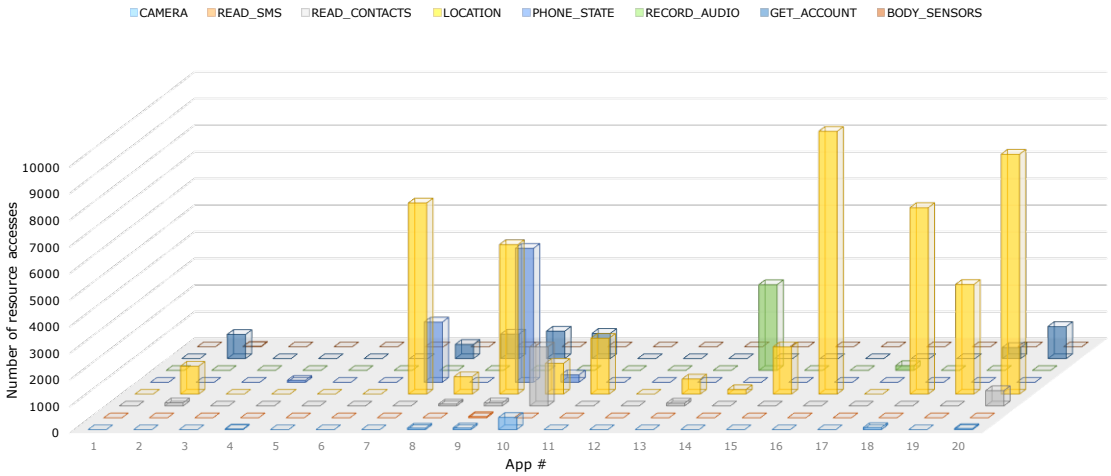
To analyze the behavior of the 20 apps within the category *Health & Fitness*, we installed the implemented monitoring tool on six Android smartphones. We also installed all the 20 apps on each of them. Next, we conducted an experiment in two phases, while the monitoring tool was running in the background the whole time (i.e. it was monitoring the privacy behavior of the apps). In the first phase (ranging from October 01, 2018 to October 10, 2018), we did not open the apps, and therefore, we never interacted with the smartphones during this time. In the second phase (ranging from October 11, 2018 to October 20, 2018), we once opened them and made few interactions (e.g. making account if needed) and let them to be executed in the background (without any further interaction). Afterwards, we collected and analyzed the data generated by the monitoring tool. In total, nine sensitive resources were accessed by the apps. The results of the analysis for each app and the resources are shown by Fig. 9(a) (Phase I: passive phase) and Fig. 9(b) (Phase II: active phase).

There is a significant increase in the number of resource accesses from Phase I to Phase II. In our case study we marked app privacy deviated behavior or privacy misbehavior if the app: (1) accessed resources besides the storage during the passive phase, or (2) accessed resources in the active phase which are not required for the app functionality. For instance, a health-based app traditionally needs access to BODY_SENSOR. The accesses to READ_STORAGE are not surprising during the two phases since the smartphones were not completely turned off and all apps could read and write files placed on the external storage (e.g. cache files). However, five apps accessed CAMERA in the passive phase (apps 8, 9, 10, 18 and 20). These accesses are not privacy-friendly, since the user does not know that the app currently accesses the camera. Furthermore, READ_CONTACTS were accessed by six apps during the passive phase. In general, such accesses to the contacts should not be done by apps. In our case the apps are health-based, where it is not clear why they need access to the user's contacts. We assume accesses to LOCATION must not be a problem for the user's privacy in general since this permission is not extremely unrelated to the functionality of the apps. However, the majority of them accessed location in passive phase. This is problematic from a privacy point of view since the apps are not location-based and, therefore, they do not need the location information to function while they are running in the passive mode. PHONE_STATE is an interesting data resource since the respective information is highly privacy sensitive. This permission enables an invasive party to gain access to sensitive resources such as phone number, cellular network information, outgoing call information, etc. The only relevant reason to access this permission is to stop the app when there is an ongoing call, however, we did not use SIM card on the devices, therefore, there is no obvious reason of such resource access. Also, there is a huge number of accesses to this resource during the passive phase which is surprising. This also happened to other sensitive resources such as RECORD_AUDIO, READ_SMS, etc. We also observed that many of these resource accesses happened when: (1) the devices were in horizontal orientation, (2) the devices' screen were off, (3) the proximity sensor indicated that there is no nearby object and (4) the app were either in INACTIVE mode or BACKGROUND mode. One interesting observation regarding this case study, which of course needs more extensive analysis, is the positive correlation between the results obtained from the user reviews analysis (Table 6) and the results obtained from the behavior analysis (Figure 9). We inferred that some of the apps that did not behave nicely in our case study (in terms of accessing sensitive resources without any transparent reason), were mostly reported by the users and were assigned several privacy threats by *MARS*.

Network Output Traffic Analysis. This step was important as we were not only interested to detect which personal resources were *accessed*, but also which ones were *transferred* to remote servers. We exploited the work proposed in [50] which detects the transmission of personal sensitive information to remote servers using on-device packet-level monitoring. During our 20-day experiment, the network output traffic analyzer was running on each individual Android device and was monitoring whether they transmit any sensitive information to external servers or not (for both passive and active phases). Basically, we detected three kinds of data transmission to external servers: (1) Data transmission to app server(s); (2) Data transmission to third-party server(s); (3) Data transmission to unknown/advertising server(s). We define a privacy leak as a process by which users' sensitive data is transmitted to remote servers without users' knowledge or consent. Under some circumstances, transferring users' personal data to remote servers is justifiable because of certain operations (e.g. introducing the device to a wireless network). However, this gets problematic when users' data is transmitted to third-party, advertising and analytics servers without users' awareness. Therefore, we only show the results regarding the data transmission to third-party server(s) and unknown/advertising server(s). Table 7 shows a detailed overview of the data types including Device ID, IMEI, Email, Location, Serial Number, MAC Address and Advertiser ID (a unique identifier being used for advertising purposes) transferred to remote servers by apps within the category *Health & Fitness*. As it is clear, all the apps (except one) sent at least one sensitive data type to external servers. This is quite concerning as we realized that many of these transmissions were done when we



(a)



(b)

Figure 9: Resource access frequencies of health-based apps: (a) passive phase, and (b) active phase.

were not interacting with the devices (passive phase). Not surprisingly, the most common transmitted data types are Location and Advertiser ID.

Investigating the past research [51, 52, 53] on the security and privacy issues of health-based apps in terms of data transmission to remote servers, we observed that our findings complement similar results in several ways: (1) We provide more fine-grained information concerning the leaked data types and their exact frequency; (2) We distinguish third-party servers from unknown/advertising servers to better understand the actual destination of privacy leaks; (3) We found out that some of the apps transferred sensitive data to locations outside the EU, including Canada, USA, China, Taiwan, etc. We checked the privacy policies of the aforementioned apps, and surprisingly, they failed to address how they deal with third-country data sharing protection practices (e.g. the implementation of the EU-US Privacy Shield or specific cross border data transfer agreements) which are strongly emphasized by the GDPR (Art. 13 (1f), 14 (1f), Art. 44, Art. 45, Art. 47).

4.4.3. Synthesis of the Analyses

In Table 8, we demonstrate the synthesis of all the analyses that we did regarding the user reviews, behavior and network output traffic of all the health-based apps in our data set.

The number in each cell of *App Behavior Analysis* column shows the number of times that each app accessed a certain permission. Similarly, the values in each cell of *Network Output Traffic Analysis* column indicates the number of times that each app transferred a certain data type to remote servers. Also, *User Review (Threat) Analysis* column illustrates the threats identified by *MARS* corresponding to each app (shown by ✓).

We analyzed the correlation between different analyses. The question that we were interested to answer was: *Are app behavior and app network output traffic transmission correlated*

Table 7: Data types transmitted to 3rd-party servers (3P Servers(s)) and unknown/advertising servers (U/A Server(s)) by apps within *Health & Fitness* category.

No.	AppID	Leaked Data Type	# Leaks to 3P Server(s)	# Leaks to U/A Server(s)
1	com.sec.android.app.shealth	Ad ID	105	43
2	com.google.android.apps...	IMEI, Location, Ad ID	340	116
3	com.sillens.shapeupclub	Ad ID	9	15
4	cc.pacer.androidapp	Serial Number, MAC Address, Ad ID	77	18
5	com.myfitnesspal.android	None	–	–
6	pedometer.steptracker...	Ad ID	38	82
7	com.stt.android	Device ID, IMEI, Email, Location, Ad ID	173	424
8	com.fitnesskeeper...	Email, Location, MAC Address, Ad ID	249	81
9	com.fitbit.FitbitMobile	Email, Location, Serial Number, MAC Address	758	104
10	com.nike.plusgps	MAC Address	41	33
11	com.runtastic.android	Location	5	2
12	com.popularapp.sevenmins	IMEI, Ad ID	30	46
13	com.popularapp.thirtyday...	Location	11	72
14	si.modula.android.instan...	Location, Ad ID	90	4
15	com.playsimple.fitnessapp	Ad ID	9	78
16	com.mapmyrun.android2	Ad ID	18	21
17	com.macropinch.hydra..	Ad ID	5	13
18	com.fitness22.running	Location, Ad ID	196	301
19	com.endomondo.android	Email, MAC Address	96	1
20	comm.cchong.BloodAssistant	Location, MAC Address, Ad ID	210	44

with user reviews analysis?. To this end, we performed a Pearson correlation test in SPSS [54] to examine whether there are relationships between app behavior, network output traffic and user reviews analyses. The Pearson correlation was used because the data met the assumptions of parametric test. Such correlation analysis is used on the nominal data, and it measures the strength of the relationship between different variables. We found a positive correlation between accessing to LOCATION permission and transferring location information to remote servers (r_p = .664, p < .001). There was also a positive significant correlation between accessing to LOCATION permission and transferring location information and device's ad ID to remote servers (r_p = .670, p < .002) resulting in lots of targeted ads complaints (T5). This is of course interesting, and to some extent concerning as the results reveal that apps accessing LOCATION permission in an excessive manner are likely to transfer device's location and device's ad ID to remote servers. Additionally, we found a negative correlation between accessing to CAMERA, READ_CONTACTS, PHONE_STATE ($P1, P3, P5$) and spam complaints (T6) in the user reviews ($r_p = -.460$, $p < .048$). On the contrary, we found a positive significant correlation between accessing to CAMERA, READ_CONTACTS, PHONE_STATE and general privacy and security complaints (T7) (r_p = .535, p < .02). Such complaints already contain a large number of reviews complaining about over-privileged apps. Such apps are known because of requesting and accessing irrelevant permissions to their proper functionality. This is also highly connected to the Principle of Least Privilege (PoLP) which was first proposed as a design principle by Saltzer and Schroeder [55]. According to the PoLP, "Every program and every user of the system should operate using the least set of privileges necessary to complete the job." Clearly, this principle is directly connected to "data minimization" (Art. 5 - 1(c) GDPR) principle, as we observed some apps accessing sensitive permissions which are irrelevant to their proper functionality. Also, the need of requesting and accessing such sensitive permissions was unclear in their privacy policy texts. We also had a closer look at the intra-threats relations, Table 9 shows the results of correlation analysis between each individual analyzed threat after user reviews mining. Looking at the relations between each individual threat, we observed a positive correlation between T1 (r_p = .504, p < .023), T4 (r_p = .601, p < .005) and T7. As for T7, a negative correlation could be also observed with T6 ($r_p = -.596$, p < .006). Furthermore, our analysis showed that there is a positive correlation between T3 and T4 (r_p = .471, p < .036), concluding that apps receiving complaints about unauthorized charges are likely to get complaints related to unintended data disclosure.

4.5. Users' Reaction to MARS: A Short Survey

We were interested: (1) to design and implement user interfaces for the sake of risk communication and (2) to analyze the reactions of real smartphone users to our transparency enhancing approach. Therefore, once all the user reviews in our data set were classified, we proceeded to aggregate the reviews to the app level within a category. To this end, we investigated the previous work [56, 57, 58, 59, 60, 61] on effective ways of privacy notices and indicators communication. Finally, we characterized our own interface design requirements as shown in Table 10.

As the main objective is to communicate to the user how much privacy is perceived to be an issue of the app (as reported by other users), we designed dedicated icons for each privacy threat. This is mainly because we decided to ease the understanding of each individual threat for the user. As a result, the users would be able to know what is the threat about in a glance,

Table 8: Results of behavior, network output traffic and user reviews analyses regarding *Health & Fitness* app category: Permissions are shown by P where P1: `CAMERA`, P2: `READ_SMS`, P3: `READ_CONTACTS`, P4: `LOCATION`, P5: `PHONE_STATE`, P6: `RECORD_AUDIO`, P7: `GET_ACCOUNT` and P8: `BODY_SENSOR`. Data leak types are shown by L where L1: Device ID, L2: IMEI, L3: Email, L4: Location, L5: Serial Number, L6: MAC Address and L7: Ad ID. Threats are shown by T where T1: Tracking & Spyware, T2: Phishing, T3: Unauthorized Charges, T4: Unintended Data Disclosure, T5: Targeted Ads, T6: Spam and T7: General.

	App Behavior Analysis								Network Output Traffic Analysis							User Review (Threat) Analysis						
App	P1	P2	P3	P4	P5	P6	P7	P8	L1	L2	L3	L4	L5	L6	L7	T1	T2	T3	T4	T5	T6	T7
1	–	–	–	–	–	–	–	–	–	–	–	–	–	–	148	✓	×	×	×	✓	✓	×
2	–	–	181	1082	–	–	1032	17	–	15	–	333	–	–	108	×	×	×	×	✓	✓	×
3	–	–	–	–	–	–	–	–	–	–	–	–	–	–	24	×	×	×	×	×	×	✓
4	27	–	–	–	88	–	–	–	–	–	–	–	10	33	52	×	✓	×	×	✓	×	×
5	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	×	×	×	×	✓	×	×
6	–	–	–	–	–	–	–	–	–	–	–	–	–	–	120	×	×	×	×	✓	✓	×
7	–	–	–	8161	2543	–	628	–	101	42	43	215	–	–	196	×	✓	✓	✓	✓	✓	×
8	83	–	125	839	–	–	1329	–	–	–	118	176	–	12	24	✓	✓	✓	✓	✓	✓	✓
9	117	63	146	6263	5561	–	1307	–	–	–	208	448	39	245	–	✓	×	✓	✓	✓	✓	✓
10	545	–	2869	1558	368	5	1310	–	–	–	–	–	–	74	–	×	×	×	×	✓	×	✓
11	–	–	–	2313	–	–	–	–	–	–	–	7	–	–	–	✓	×	✓	✓	✓	×	✓
12	–	–	–	–	–	–	34	–	–	27	–	–	–	–	49	×	×	×	×	✓	✓	×
13	–	–	140	683	–	–	–	–	–	–	–	83	–	–	–	×	×	×	×	✓	✓	×
14	–	–	–	268	–	3772	–	–	–	–	–	89	–	–	5	×	✓	✓	✓	✓	×	✓
15	–	–	–	2187	–	–	–	–	–	–	–	–	–	–	87	×	×	×	×	✓	✓	×
16	–	–	–	13644	–	–	–	–	–	–	–	–	–	–	39	×	×	✓	×	×	✓	×
17	–	–	–	–	–	217	–	–	–	–	–	–	–	–	18	✓	✓	×	✓	✓	×	✓
18	95	–	–	7672	–	–	–	–	–	–	–	362	–	–	135	×	×	×	✓	✓	×	✓
19	–	–	157	4683	–	–	518	–	–	–	96	–	–	1	–	×	×	✓	×	✓	✓	×
20	58	–	718	10533	–	–	1659	–	–	–	–	138	–	15	101	✓	✓	✓	×	✓	×	✓

Table 9: Results of correlation analysis between different analyzed threats.

T	T1	T2	T3	T4	T5	T6	T7
T1 r_p	1	.286	.356	.435	.218	-.066	.504
p	–	.222	.123	.055	.355	.783	.023
T2 r_p	.286	1	.356	.435	.218	-.285	.285
p	.222	–	.123	.055	.355	.223	.223
T3 r_p	.356	.356	1	.471	-.068	.123	.287
p	.123	.123	–	.036	.776	.605	.220
T4 r_p	.435	.435	.471	1	.245	-.179	.601
p	.055	.055	.036	–	.299	.450	.005
T5 r_p	.218	.218	-.068	.245	1	.034	-.034
p	.355	.355	.776	.299	–	.888	.888
T6 r_p	-.066	-.285	.123	-.179	.034	1	-.596
p	.783	.223	.605	.450	.888	–	.006
T7 r_p	.504	.285	.287	.601	-.034	-.596	1
p	.023	.223	.220	.005	.888	.006	–

Table 10: User interface design requirements.

#	Requirement	Rationale
R1	Informative	The interface should support sufficient information regarding the instructions for using different screens of the proposed prototype (to make it easy to learn)
R2	Response Time	It is defined as the time that it takes for the user to send/receive any reaction to/from the interface.
R3	Tedium	The interaction between user and user interface should be kept in the maximum possible level of attractiveness (avoiding non-informative privacy indicators full of legal and technical descriptions leading to frustration).
R4	Ambiguity	The interface should be straightforward and not confusing for the user (being understandable to a wide range of users regardless of age, knowledge, education, etc.).
R5	Attractiveness	The interface should be attractive. Users do not want to follow what they do not like. Therefore, the interface should trigger the users' attention and interest considering different sensitivity levels by colors and icons.
R6	Fear	The interface should be reliable and it should not impose fear on the user (e.g. by showing superficial, unrealistic and too direct information).

considering a binary risk level indicated by colors ranging from red (extreme warning) in case of a threat is flagged to green (calming). As a further step, if the users would be interested to know more about each privacy threat in detail and to read the relevant published reviews for each threat, they can simply click on each dedicated icon. Fig. 10 demonstrates the different screens regarding the proposed interfaces.

To examine the reaction of real users to *MARS* including its generated results and user interface design, we performed a short scale user study. We hired 41 participants through a random selection model at our university campus and advertising on social networks (e.g. Facebook). In order to participate in the user study, the participants must be over 18 years. To reduce potential biases, we requested for participants without advanced knowledge in computer science and IT related areas. Among 41 study participants, the majority (56.2%) were between 18-24 years old. Most of the respondents held higher education, either bachelor's (46.3%) or master's degree and higher (12.2%). The detailed demographics are presented in Table 11.

31 (75.6%) participants indicated that the reputation of an

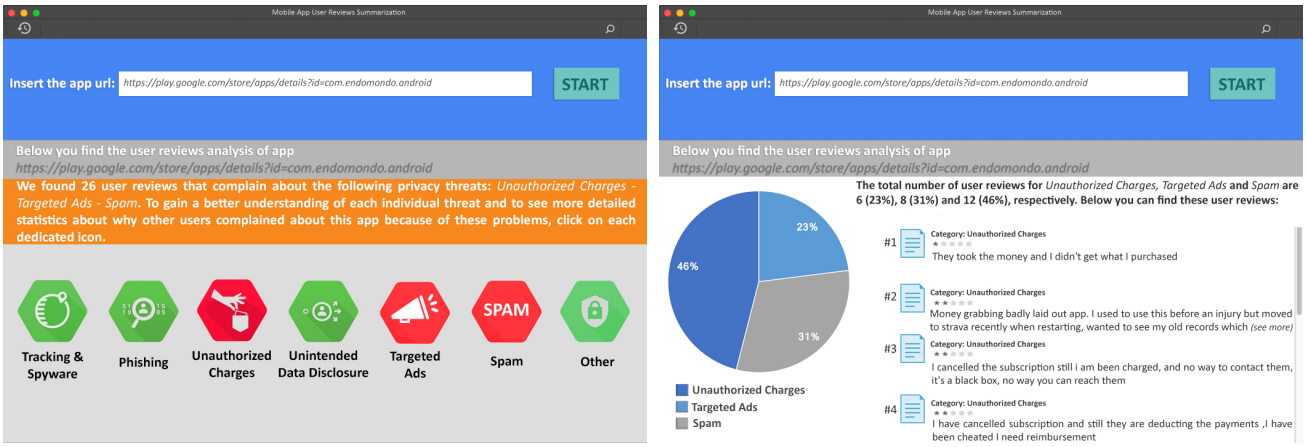


Figure 10: The proposed interface for *MARS*: (a) summarized privacy threats and their respective icons, (b) privacy relevant user reviews associated to each identified privacy threat.

Table 11: Participants demographics.

Demographic	Group	N	%
Age	18-24	23	56.2
	25-34	12	29.2
	35 or older	6	14.6
Gender	Female	26	63.4
	Male	15	36.6
Education	High school	6	14.6
	Some college	11	26.8
	Bachelors degree	19	46.3
	Masters degree or higher	5	12.2
IT experience	Not at all	4	9.7
	Trivial	21	51.2
	Moderate	12	29.2
	A lot	4	9.7

app (e.g. rating on the Google Play Store, number of downloads, number of associated user reviews, etc.) is important in their decision to give their personal information to it, while six (14.6%) participants only stated it is slightly important. In addition, only five (12.2%) participants stated that they cautiously read user reviews before downloading a new smartphone app, while 29 (70.7%) said they never read it and seven (17%) said they read it rarely. The most given reason was, that some participants said they only go through the latest reviews as reading all the reviews is very time consuming for them. Generally, the design of the user interface was appreciated. On a scale of 0 to 5 with 5 being the best design, a mean value of $3,9$ could be reached. On the questions whether they would actually use such a tool, $82,5\%$ answered with the highest score. The mean answer (again on a scale from 0 to 5) was $4,4$. The participants also highlighted that they are interested in using such a tool to save time and to obtain a good overview of the degree of privacy protection of apps by having an easy-to-use system that enables them to quickly analyze a large amount of reviews. Several participants also indicated that they would not be capable of assessing the privacy protection themselves, and therefore, using *MARS* helps them to get benefit from crowdsource's knowledge. Moreover, it was appreciated that such an approach increases transparency, especially for the inexperienced users, who want to install a certain app for the first time. When asked how to improve the functionality of our tool, it was suggested to calculate an overall score for an app based on the analysis of its privacy and security relevant user reviews and to provide more information on how each defined threat can menace users' privacy. Participants also proposed to make recommendations on alternative apps, if the evaluation does not yield a good result.

4.6. Discussion and Key Insights

We are all surrounded by mobile devices, especially smartphones and people are heavily relied on such devices. Consequently, mobile apps are now indispensable parts of our daily life. However, privacy and security have become challenging and controversial topics in the area of mobile apps as they are highly dependent on users' personal information to provide functionality. App markets already started to improve their privacy and security mechanisms, e.g. Google started to filter out privacy-unfriendly apps (e.g. malware) from its app market (790K apps were removed during a two year period from 2015 to 2017) [62]. Nevertheless, we strongly believe that the area of mining smartphone users' privacy perception is not well-explored and calls for action need to be seriously adopted.

Needless to say, our results showed that there is a correlation between what the mobile apps are doing in reality and how they are perceived from users' perspective. This confirms that models on privacy and data protection practices of mobile apps are problematic and regulations become necessary when users do not have the chance to get correct and full information about data collection of apps. Thus, we argue that app developers should carefully clarify the needs of requesting sensitive permissions. One way is to benefit from the proposed approach in this paper to improve the data protection quality of their apps. We expect that in the future this could positively impact developers; once their apps are being compared to others regard-

ing the privacy aspects, they would be encouraged to take action and provide more privacy-friendly apps.

By simplifying the access to privacy relevant user reviews, the presently proposed approach makes it easier for developers to understand more quickly the privacy and security maintenance tasks to apply, and, consequently, to enhance the privacy and security properties of their apps. Not only app developers, but also mobile users can benefit from our system. Thanks to the proposed summarization interfaces, the users can simply analyze app user reviews for finding privacy relevant information. Hence, our proposed approach can be either used as a standalone or embedded in smartphone ecosystems leading to efficient and time-saving analysis of user reviews. Further, users usually uninstall an app found as privacy/security invasive without precautionary broadcast (e.g. publishing reviews). Correspondingly, they need more incentives to write and publish reviews regarding their experience in dealing with privacy and security issues. Easy to use, automated and fast procedures enabling users for simplified review writing and publishing would be quite effective to revamp this situation. In addition, users easily get disappointed once they realize that their personal information is accessed without their consent [63]. App developers' responsiveness would amend this situation to a certain degree. Hence, the developers should not stop or ignore communication with users. Also, a record of privacy and security enhancements adopted by the developers should be at hand to users, enabling them to track the improvements. This would ideally fulfill the transparency principle as one of the important data protection aspects. As another crucial factor regarding the importance of users' privacy perception analysis, we observed interesting correlations between what apps are doing in reality and what users' perceive about them, which in turn, shows how *MARS* can be helpful for both developers and users to potentially gain knowledge about the privacy aspects of apps.

4.7. Limitations and Future Work

In this study, we have only focused on ten app categories (20 apps per category). A more extensive study could provide more insights on the validity of our results. However, as the top ten apps are usually judged as a measure of the app downloads on app markets [64], we may argue that our findings might exert to the apps with the greatest impact on the users. It is worth mentioning, that *MARS* is an ex-ante transparency tool that needs data (user reviews) to function. Accordingly, it is well performed on apps with high number of user reviews (popular apps), thus, under circumstances that less popular apps have fewer reviews, it might not perform nicely in terms of performance. However, such limitation can be tolerated as one of the main targets of *MARS* is to provide ex-ante transparency through the analysis of already published user reviews. Additionally, the overall usefulness of such a transparency enhancing system is dependent on both (1) how well it understands the user reviews and (2) how truthful those user reviews are. Our proposed approach mainly covers the former, and still there is little discussion on the accuracy, veracity, and clarity of the reviews that is generally out of our control. Our system has two main purposes targeting both app developers and users. We argue that app developers can benefit from our analysis by understanding why users complain about certain privacy and security aspects of their apps. This is a very important point as investigation of past research revealed that it is very likely that bad privacy and security practices in mobile apps only happen because of developers' lack of knowledge in API usage or wrong definition of permission requests. As a result, our system can help those app developers who unintentionally violate users' privacy and want to improve the privacy and security properties of their apps. Thus, it will enable app developers to quickly check the real users' experience in dealing with privacy and security violations, and accordingly, they will be able to fix those issues. In this regard, the existing of untrustworthy reviews cannot limit the applicability of our system, as we assume developers are experts and they are able to distinguish *fake* reviews from trustworthy ones to react accordingly. However, when it comes to the second target group of our system (users), untrustworthiness of reviews might be influential as distinguishing fake reviews from genuine reviews is not an easy task for normal users. To possibly reduce this risk, *MARS* can be complemented with our proposed app behavior monitoring tool. This can be done to prevent the negative impact that fake reviews might have in the overall performance of *MARS*. As a result, once *MARS* is accompanied by our proposed behavior monitoring tool, we can expect more coherent and reliable results. It is worth mentioning that, we took the following considerations into account while implementing *MARS* to minimize the potential negative effect of untrustworthy reviews on our results concerning the second target group (users): (1) Same user ID, different apps: As each user review in our data set is accompanied by a user ID, we can detect those users who have published reviews for multiple apps with a same ID. In this regard, a certain user review associated with multiple apps within a certain category (with a same user ID), is likely to be an untrustworthy review [65, 66]. Thus, such reviews have been ignored; (2) Review length: The length of reviews plays a critical role in the overall analysis. For instance, a spammer oftentimes tends to leave short reviews (one or few words) to just negatively/positively influence the overall rating of an app. We do only analysis those reviews that have more than 5 words as an indicator of informativeness degree [67, 68, 69]. Further, our results showed that there is still a low number of overall privacy relevant user reviews (compared to other topics), thus, as a future work we aim to provide a tool that will ease users' understanding towards the privacy behavior of their installed apps, and a reporting tool to support users in a semi-automatic elaboration of app privacy-related reports. Moreover, we believe more extensive user studies can supply further insights on privacy views to increase the smartphone users' privacy awareness. In addition, ranking algorithms could be initialized to mathematically score apps based on invasiveness level of their reviews and their behavior in reality. This way the user can be provided with tailored means to become informed and to make decisions easily, which are, ultimately, important purposes of our system.

5. Conclusion

In this paper, we proposed an approach for the automatic analysis of users' privacy perception embedded in app user reviews. Our goal was to provide insights regarding mining app user reviews and the relation between privacy relevant user reviews analysis and apps' real behavior. Therefore, we proposed *MARS* based on ML, NLP and sentiment analysis techniques as a summarization tool that eases the process of understanding privacy relevant statements hidden in app user reviews, and to support developers to effectively examine such statements. Our findings revealed the following interesting facts regarding the performance of *MARS*, mobile app privacy practice and users' privacy concern and perception. In terms of performance, *MARS* outperformed similar relevant works. We have further found user reviews on app markets to be an important source that provide granular information about apps' privacy behavior. By looking into user reviews, we showed the possibility of inferring apps' potential privacy threats that would help developers to address privacy and security issues of their apps. Also, it enables users to compare apps in terms of privacy aspects through the proposed summarization interfaces. We also investigated the relation between users' privacy perception and what mobile apps are doing in reality through doing a behavior analysis jointly with network output traffic monitoring that signaled a positive correlation.

References

[1] Number of available applications in the google play store from december 2009 to june 2017, https://www.statista.com/statistics/266210/number-of-available-applications-in-the-google-play-store/, accessed august 8, 2018 (2017).

[2] Google announces over 2 billion monthly active devices on android, https://www.theverge.com/2017/5/17/15654454/android-reaches-2-billion-monthly-active-users/, accessed august 8, 2018 (2017).

[3] M. Green, M. Smith, Developers are not the enemy!: The need for usable security apis, IEEE Security Privacy 14 (5) (2016) 40–46. doi:10.1109/MSP.2016.111.

[4] G. Lackermair, D. Kailer, K. Kanmaz, Importance of online product reviews from a consumer's perspective, Advances in Economics and Business 1 (1) (2013) 1–5.

[5] P. Gilbert, B. G. Chun, L. Cox, J. Jung, Automating privacy testing of smartphone applications, Tech. Rep. CS-2011-02, Duke University (2011).

[6] A. Beresford, A. Rice, N. Sohan, Mockdroid: trading privacy for application functionality on smartphones, in: the Proceedings of the 12th Workshop on Mobile Computing Systems and Applications, Phoenix, Arizona, USA, 2011, pp. 49–54.

[7] Y. Zhou, X. Zhang, X. Jiang, V. W. Freech, Taming information-stealing smartphone applications (on android), in: the Proceedings of the 4th International Conference on Trust and Trustworthy Computing, Pittsburgh, PA, USA, 2011, pp. 39–107.

[8] P. Pearce, A. P. Felt, G. Nunez, D. Wagner, Addroid: Privilege separation for applications and advertisers in android, in: Proceedings of the 7th ACM Symposium on Information, Computer and Communications Security, ASIACCS '12, ACM, New York, NY, USA, 2012, pp. 71–72. doi:10.1145/2414456.2414498.
URL http://doi.acm.org/10.1145/2414456.2414498

[9] V. F. Taylor, I. Martinovic, Securank: Starving permission-hungry apps using contextual permission analysis, in: Proceedings of the 6th Workshop on Security and Privacy in Smartphones and Mobile Devices, SPSM '16, ACM, New York, NY, USA, 2016, pp. 43–52. doi:10.1145/2994459.2994474.
URL http://doi.acm.org/10.1145/2994459.2994474

[10] W. Enck, M. Ongtang, P. Mcdaniel, On lightweight mobile phone application certification, in: the Proceedings of the the 16th ACM Conference on Computer and Communications Security, Chicago, Illinois, USA, 2009, pp. 235–245.

[11] W. Enck, D. Octeau, P. McDaniel, S. Chaudhuri, A study of android application security, in: the Proceedings of the the 20th USENIX Conference on Security, San Francisco, CA, USA, 2011, pp. 21–21.

[12] W. Enck, P. Gilbert, B. Chun, L. P. Cox, J. Jung, P. McDaniel, A. N. Sheth, Taintdroid: an information-flow tracking system for realtime privacy monitoring on smartphones, in: the Proceedings of the the 9th ACM USENIX Conference on Operating Systems Design and Implementation, Vancouver, BC, Canada, 2010, pp. 393–407.

[13] S. M. Habib, N. Alexopoulos, M. M. Islam, J. Heider, S. Marsh, M. Müehlhäeuser, Trust4app: Automating trustworthiness assessment of mobile applications, in: 2018 17th IEEE International Conference On Trust, Security And Privacy In Computing And Communications/12th IEEE International Conference On Big Data Science And Engineering (TrustCom/BigDataSE), IEEE, 2018, pp. 124–135.

[14] P. Wijesekera, A. Baokar, A. Hosseini, S. Egelman, D. Wagner, K. Beznosov, Android permissions remystified: A field study on contextual integrity, in: Proceedings of the 24th USENIX Conference on Security Symposium, SEC'15, USENIX Association, Berkeley, CA, USA, 2015, pp. 499–514.
URL http://dl.acm.org/citation.cfm?id=2831143.2831175

[15] B. Fu, J. Lin, L. Li, C. Faloutsos, J. Hong, N. Sadeh, Why people hate your app: Making sense of user feedback in a mobile app store, in: Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '13, ACM, New York, NY, USA, 2013, pp. 1276–1284. doi:10.1145/2487575.2488202.
URL http://doi.acm.org/10.1145/2487575.2488202

[16] C. Iacob, R. Harrison, Retrieving and analyzing mobile apps feature requests from online reviews, in: 2013 10th Working Conference on Mining Software Repositories (MSR), 2013, pp. 41–44. doi:10.1109/MSR.2013.6624001.

[17] L. V. Galvis Carreño, K. Winbladh, Analysis of user comments: An approach for software requirements evolution, in: Proceedings of the 2013 International Conference on Software Engineering, ICSE '13, IEEE Press, Piscataway, NJ, USA, 2013, pp. 582–591.
URL http://dl.acm.org/citation.cfm?id=2486788.2486865

[18] J. Oh, D. Kim, U. Lee, J.-G. Lee, J. Song, Facilitating developer-user interactions with mobile app review digests, in: CHI '13 Extended Abstracts on Human Factors in Computing Systems, CHI EA '13, ACM, New York, NY, USA, 2013, pp. 1809–1814. doi:10.1145/2468356.2468681.
URL http://doi.acm.org/10.1145/2468356.2468681

[19] D. Pagano, W. Maalej, User feedback in the appstore: An empirical study, in: 2013 21st IEEE International Requirements Engineering Conference (RE), 2013, pp. 125–134. doi:10.1109/RE.2013.6636712.

[20] E. Guzman, W. Maalej, How do users like this feature? a fine grained sentiment analysis of app reviews, in: 2014 IEEE 22nd International Requirements Engineering Conference (RE), 2014, pp. 153–162. doi:10.1109/RE.2014.6912257.

[21] S. C. H. H. X. X. N. N. Chen, J. Lin, B. Z. Nanyang, Ar-miner: Mining informative reviews for developers from mobile app marketplace, in: in the Proceedings of the 36th International Conference on Software Engineering, India, 2014, pp. 767–778.

[22] L. Cen, L. Si, N. Li, H. Jin, User comment analysis for android apps and cspi detection with comment expansion, in: the Proceedings of the 1st International Workshop on Privacy-Preserving IR (PIR), Gold Coast, Australia, 2014, pp. 25–30.

[23] D. Kong, L. Cen, H. Jin, Autoreb: Automatically understanding the review-to-behavior fidelity in android applications, in: the Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security, Denver, Colorado, USA, 2015, pp. 530–541.

[24] S. Panichella, A. Di Sorbo, E. Guzman, C. A. Visaggio, G. Canfora, H. C. Gall, How can i improve my app? classifying user reviews for software maintenance and evolution, in: Proceedings of the 2015 IEEE International Conference on Software Maintenance and Evolution (ICSME), ICSME '15, IEEE Computer Society, Washington, DC, USA, 2015, pp. 281–290. doi:10.1109/ICSM.2015.7332474.
URL http://dx.doi.org/10.1109/ICSM.2015.7332474

[25] H. Khalid, E. Shihab, M. Nagappan, A. E. Hassan, What do mobile app users complain about?, IEEE Software 32 (3) (2015) 70–77. doi:10.1109/MS.2014.50.

[26] S. Mcilroy, N. Ali, H. Khalid, A. E. Hassan, Analyzing and automatically labelling the types of user issues that are raised in mobile app reviews, Empirical Softw. Engg. 21 (3) (2016) 1067–1106. doi:10.1007/s10664-015-9375-7.
URL http://dx.doi.org/10.1007/s10664-015-9375-7

[27] A. Ciurumelea, A. Schaufelbuhl, S. Panichella, H. C. Gall, Analyzing reviews and code of mobile apps for better release planning, in: 2017 IEEE 24th International Conference on Software Analysis, Evolution and Reengineering (SANER), 2017, pp. 91–102. doi:10.1109/SANER.2017.7884612.

[28] D. C. Nguyen, E. Derr, M. Backes, S. Bugiel, Short text, large effect: Measuring the impact of user reviews on android app security & privacy, in: 2019 IEEE Symposium on Security and Privacy (SP), Vol. 00, pp. 155–169. doi:10.1109/SP.2019.00012.
URL doi.ieeecomputersociety.org/10.1109/SP.2019.00012

[29] Google play scraper, https://github.com/facundoolano/google-play-scraper/.

[30] Natural language toolkit," https://www.nltk.org/.

[31] C. J. Hutto, E. E. Gilbert, Vader: A parsimonious rule-based model for sentiment analysis of social media text, in: the Proceedings of the 8th International Conference on Weblogs and Social Media, Ann Arbor, MI, USA, 2014, pp. 530–541.

[32] G. Hogben, M. Dekker, Smartphones: Information security risks, opportunities and recommendations for users, Tech. rep., ENISA report (2010).

[33] Mobile top 10 2016-top 10, https://www.owasp.org/index.php/mobile_top_10_2016-top_10/, accessed august 8, 2018 (2016).

[34] L. Marinos, Enisa threat taxonomy: A tool for structuring threat information, Tech. rep., ENISA report (2016).

[35] Eu general data protection regulation, https://eur-lex.europa.eu/legal-content/en/txt/html/?uri=celex:32016r0679, accessed august 8, 2018 (2016).

[36] Eu-u.s. privacy shield, https://iapp.org/resources/article/eu-u-s-privacy-shield-full-text/, accessed august 12, 2018 (2016).

[37] J. Pennington, R. Socher, C. D. Manning, Glove: Global vectors for word representation, in: the Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP), Doha, Qatar, 2014, pp. 1532–1543.

[38] D. Jurafsky, J. H. Martin, Speech and language processing: An introduction to natural language processing, computational linguistics, and speech recognition, Pearson, 2000.

[39] F. Pedregosa, G. Varoquaux, A. Gramfort, B. T. V. Michel, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, J. Passos, D. Cournapeau, M. Brucher, M. Perrot, E. Duchesnay, Scikit-learn: Machine learning in python, Journal of Machine Learning Research 12 (2011) 2825–2830.

[40] E. Alpaydin, Introduction to machine learning, MIT Press, 2014.

[41] J. T. Pohlman, D. W. Leitner, A comparison of ordinary least squares and logistic regression, The Ohio Journal of Science 103 (2003) 118–125.

[42] B. Martínez-Pérez, I. De La Torre-Díez, M. López-Coronado, Privacy and security in mobile health apps: A review and recommendations, J. Med. Syst. 39 (1) (2015) 1–8.

[43] M. Plachkinova, S. Andres, S. Chatterjee, A taxonomy of mhealth apps – security and privacy concerns, in: 2015 48th HICSS, 2015, pp. 3187–3196.

[44] L. Hutton, B. A. Price, R. Kelly, C. McCormick, A. K. Bandara, T. Hatzakis, M. Meadows, B. Nuseibeh, Assessing the privacy of mhealth apps for self-tracking: Heuristic evaluation approach, JMIR Mhealth Uhealth 6 (10) (2018) e185. doi:10.2196/mhealth.9217.

[45] A. Papageorgiou, M. Strigkos, E. Politou, E. Alepis, A. Solanas, C. Patsakis, Security and privacy analysis of mobile health applications: The alarming state of practice, IEEE Access 6 (2018) 9390–9403. doi:10.1109/ACCESS.2018.2799522.

[46] Facebook data privacy scandal: A cheat sheet, https://www.techrepublic.com/article/facebook-data-privacy-scandal-a-cheat-sheet/, accessed jan 11, 2019 (2018).

[47] M. Hatamian, J. Serna, K. Rannenberg, B. Igler, Fair: Fuzzy alarming index rule for privacy analysis in smartphone apps, in: the Proceedings of the 14th International Conference on Trust and Privacy in Digital Business (TrustBus), Lyon, France, 2017, pp. 3–18.

[48] M. Hatamian, A. Kitkowska, J. Korunovska, S. Kirrane, "it's shocking¡': Analysing the impact and reactions to the a3: Android apps behaviour analyser, in: F. Kerschbaum, S. Paraboschi (Eds.), Data and Applications Security and Privacy XXXII, Springer International Publishing, Cham, 2018, pp. 198–215.

[49] Google removes vital privacy feature from android, claiming its release was accidental, https://www.eff.org/deeplinks/2013/12/google-removes-vital-privacy- features-android-shortly-after-adding-them/, accessed august 8, 2018 (2013).

[50] A. Shuba, A. Le, M. Gjoka, J. Varmarken, S. Langhoff, A. Markopoulou, Antmonitor: Network traffic monitoring and real-time prevention of privacy leaks in mobile devices, in: Proceedings of the 2015 Workshop on Wireless of the Students, by the Students, & for the Students, S3 '15, ACM, New York, NY, USA, 2015, pp. 25–27. doi:10.1145/2801694.2801707.
URL http://doi.acm.org/10.1145/2801694.2801707

[51] H. Dongjing, N. Muhammad, G. A. Carl, N. Klara, Security concerns in android mhealth apps, AMIA Annu Symp Proc. 2014 (2014) 645–654.

[52] K. Huckvale, J. T. Prieto, M. Tilney, P.-J. Benghozi, J. Car, Unaddressed privacy risks in accredited health and wellness apps: a cross-sectional systematic assessment, BMC Medicine 13 (1) (2015) 214. doi:10.1186/s12916-015-0444-y.
URL https://doi.org/10.1186/s12916-015-0444-y

[53] K. Knorr, D. Aspinall, M. Wolters, On the privacy, security and safety of blood pressure and diabetes apps, in: H. Federrath, D. Gollmann (Eds.), ICT Systems Security and Privacy Protection, Springer International Publishing, Cham, 2015, pp. 571–584.

[54] J. M. A. Field, Z. Field, Discovering statistics using SPSS, Sage Publications Ltd, 2013.

[55] J. H. Saltzer, M. D. Schroeder, The protection of information in computer systems, Proceedings of the IEEE 63 (9) (1975) 1278–1308. doi:10.1109/PROC.1975.9939.

[56] V. C. Conzola, M. S. Wogalter, A communication-human information processing (c-hip) approach to warning effectiveness in the workplace, Journal of Risk Research 4 (2001) 309–322.

[57] J. Angulo, S. Fischer-Hübner, T. Pulls, E. Wästlund, Usable transparency with the data track: A tool for visualizing data disclosures, in: Proceedings of the 33rd Annual ACM Conference Extended Abstracts on Human Factors in Computing Systems, CHI EA '15, ACM, New York, NY, USA, 2015, pp. 1803–1808. doi:10.1145/2702613.2732701.
URL http://doi.acm.org/10.1145/2702613.2732701

[58] F. Schaub, R. Balebako, A. L. Durity, L. F. Cranor, A design space for effective privacy notices, in: Proceedings of the Eleventh USENIX Conference on Usable Privacy and Security, SOUPS'15, USENIX Association, Berkeley, CA, USA, 2015, pp. 1–17.
URL http://dl.acm.org/citation.cfm?id=3235866.3235868

[59] L. F. Cranor, P. Guduru, M. Arjula, User interfaces for privacy agents, ACM Trans. Comput.-Hum. Interact. 13 (2) (2006) 135–178. doi:10.1145/1165734.1165735.
URL http://doi.acm.org/10.1145/1165734.1165735

[60] P. Raschke, A. Küpper, O. Drozd, S. Kirrane, Designing a GDPR-Compliant and Usable Privacy Dashboard, Springer International Publishing, Cham, 2018, pp. 221–236. doi:10.1007/978-3-319-92925-5_14.
URL https://doi.org/10.1007/978-3-319-92925-5_14

[61] P. Murmann, A. Kitkowska, P. S. Kumar, M. Hatamian, A. Ralien, J. Quintero, L. Abdullah, A. Mittos, M. Warner, A. Gutmann, D4.1 user interface requirements, Deliverable, USECON, Privacy&Us.

[62] H. Wang, H. Li, L. Li, Y. Guo, G. Xu, Why are android apps removed from google play?: A large-scale empirical study, in: Proceedings of the 15th International Conference on Mining Software Repositories, MSR '18, ACM, New York, NY, USA, 2018, pp. 231–242. doi:10.1145/3196398.3196412.
URL http://doi.acm.org/10.1145/3196398.3196412

[63] B. Liu, J. Lin, N. Sadeh, Reconciling mobile app privacy and usability on smartphones: Could user privacy profiles help?, in: Proceedings of the 23rd International Conference on World Wide Web, WWW '14, ACM, New York, NY, USA, 2014, pp. 201–212. doi:10.1145/2566486.2568035.
URL http://doi.acm.org/10.1145/2566486.2568035

[64] N. Zhong, F. Michahelles, Google play is not a long tail market: An empirical analysis of app adoption on the google play app market, in: Proceedings of the 28th Annual ACM Symposium on Applied Computing, SAC '13, ACM, New York, NY, USA, 2013, pp. 499–504. doi:10.1145/2480362.2480460.
URL http://doi.acm.org/10.1145/2480362.2480460

[65] N. Jindal, B. Liu, Opinion spam and analysis, in: Proceedings of the 2008 International Conference on Web Search and Data Mining, WSDM '08, ACM, New York, NY, USA, 2008, pp. 219–230. doi:10.1145/1341531.1341560.
URL http://doi.acm.org/10.1145/1341531.1341560

[66] Y.-R. Chen, H.-H. Chen, Opinion spammer detection in web forum, in: Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR '15, ACM, New York, NY, USA, 2015, pp. 759–762. doi:10.1145/2766462.2767766.
URL http://doi.acm.org/10.1145/2766462.2767766

[67] E.-P. Lim, V.-A. Nguyen, N. Jindal, B. Liu, H. W. Lauw, Detecting product review spammers using rating behaviors, in: Proceedings of the 19th ACM International Conference on Information and Knowledge Management, CIKM '10, ACM, New York, NY, USA, 2010, pp. 939–948. doi:10.1145/1871437.1871557.
URL http://doi.acm.org/10.1145/1871437.1871557

[68] C. L. Lai, K. Q. Xu, R. Y. K. Lau, Y. Li, L. Jing, Toward a language modeling approach for consumer review spam detection, in: 2010 IEEE 7th International Conference on E-Business Engineering, 2010, pp. 1–8. doi:10.1109/ICEBE.2010.47.

[69] Y. Lin, T. Zhu, H. Wu, J. Zhang, X. Wang, A. Zhou, Towards online anti-opinion spam: Spotting fake reviews from the review sequence, in: 2014 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM 2014), 2014, pp. 261–264. doi:10.1109/ASONAM.2014.6921594.