AN EMPIRICAL STUDY ON USER REVIEWS TARGETING MOBILE APPS' SECURITY & PRIVACY

Debjyoti Mukherjee
Department of Computer Science
University of Calgary, Canada
debjyoti.mukherje1@ucalgary.ca

Alireza Ahmadi
Department of Computer Science
University of Calgary, Canada
alireza.ahmadi1@ucalgary.ca

Maryam Vahdat Pour
Schulich School of Engineering
University of Calgary, Canada
maryam.vahdatpour@ucalgary.ca

Joel Reardon
Department of Computer Science
University of Calgary, Canada
joel.reardon@ucalgary.ca

ABSTRACT

Application markets provide a communication channel between app developers and their end-users in form of app reviews, which allow users to provide feedback about the apps. Although security and privacy in mobile apps is one of the biggest issues, it is unclear how much people are aware of these or discuss about them in reviews.
In this study, we explore the privacy and security concerns of users using reviews in the Google Play Store. For this we conducted a study by analyzing around 2.2M reviews from the top 539 apps of this Android market. We found that 0.5% of these reviews are related to the security and privacy concerns of the users. We further investigated these apps by performing dynamic analysis which provided us valuable insights into their actual behaviours. Based on the different perspectives, we categorized the apps and evaluated how the different factors influence the users' perception about the apps. It was evident from the results that the number of permissions that the apps request plays a dominant role in this matter. We also found that sending out the location can affect the users' thoughts about the app. The other factors do not directly affect the privacy and security concerns for the users.

Keywords Natural Language Processing · Google Play · App Review · Mobile Application · Security · Privacy · Dynamic Analysis

1 Introduction

Mobile application have been a part of computers for over a decade and mobile software market is the fastest growing segment in the mobile industry. With the ever-growing popularity of mobile apps, various OS providers and device vendors have launched their own application stores; Google Play and Apple's App Store are the two most popular among them. These markets distribute many apps for end-users to search, download, and purchase applications. Similar to online retail markets, end-user reviews are a key for the success of the apps. Users that have used an app can write reviews—including a 1-to-5-star rating—to express their opinion about an app and help other users to choose among similar apps.

Reviews can also be used as a direct feedback channel to app developers. Developers can find out feature suggestions, as well as usability issues, crashes, and other types of feedback about their apps. While prior research has focused on providing users with support for choosing less risky apps [1, 2] or helping them making informed decisions [3, 4, 5], there is a dearth of research related to this feedback channel. We believe that for apps to improve their security-and-privacy related behavior, feedback should be directed to developers. User reviews would seemingly form such an immediate feedback and rating channel for security and privacy related concerns from users.

This paper performs an empirical study on mobile app reviews for top free mobile applications in Google Play to explore how much the users are concerned about their security-and-privacy while using mobile apps and whether their concerns are justified. In this study, we have answered the following two research questions:
RQ 1: How much are users concerned about security and privacy while using mobile apps?
RQ 2: To what extent, users' judgment matches the actual functionality of mobile apps?

To answer RQ1, we mined the reviews and apps' details from the app store and performed supervised learning to identify the security-and-privacy related reviews. We refer to these reviews as *"Related"* reviews and observed that a considerable number of reviews are related. We also devised a method to identify if the related reviews correspond to a positive or a negative sentiment. Based on this classification, we have successfully labelled the apps.

In order to evaluate RQ2, we performed dynamic analysis to check the actual behaviour of the apps and collected various types of data like Personal Identifiable Information (PII) leaked, the different hosts connected, the type of permissions asked, etc. On the basis of these data we have again categorized the apps. Finally we have compared the categorization of apps based on reviews to new categories based on the dynamic analysis and have successfully answered the second research question. We have also used different statistical measures to evaluate our results.

In summary, the contributions of this work are the following:

- We demonstrate a way to judge how much the users of mobile apps are concerned about their privacy-and-security
- We have measured the actual behavior of the apps as it relates to privacy and could compare how much of the users' concerns are justified
- We have identified some of the key factors that users' depend on while judging any app's behavior related to privacy
- We show that reviews can be useful for identifying some types of privacy violation in mobile apps, while it may not be as effective in some other aspects

The rest of this paper is structured as follows: Section 2 provides an overview of the related work, Section 3 depicts our methodology, Section 4 provides overview of the data analyzed, Section 5 shows the results, Section 6 contains the discussion, and Section 7 has the conclusion.

2 Related Work

Android privacy, and in particular application privacy and the role of developers in the mobile ecosystem, have been studied from a variety of perspectives. In this section, we survey related works on app reviews in general and how describe privacy issue and awareness on apps using natural language. We would also take a look at app security evolution.

2.1 General App Reviews

App reviews are the primary channel through which developers receive feedback about their applications. Prior work by Pagano and Maalej [6] found that different apps receive different amount of reviews, and reviews are not easy to automatically analyze given their unstructured forms. Chen et al. [7] has shown that about one third of the user reviews are informative and focused on automatically identifying useful user reviews for developers. Existing work by Palomba et al. [8] also proposed a similar approach to support app developers in classifying feedback useful for app maintenance. Fu et al. [9] proposed a tool that analyzes user comments and ratings in mobile app markets. The approach uses regression and Latent Dirichlet Allocation (LDA) [10] models to analyze the comments' topics. In contrast, our study focuses on the connection between app reviews and the application's security and privacy concerns.

2.2 Developer Reviews

Past research has successfully mined software artifacts and connected them with the app descriptions regarding security and privacy aspects. Gorla et al. [2] proposed an approach to examine whether the applications' descriptions matches the applications' behavior. It offers a solution to cluster apps by their topics based on their description, and the usage of permission for protected APIs within each cluster. Further, Pandita et al. [11] and Qu et al. [1] proposed two systems that mine Android application descriptions and then use natural language processing (NLP) to automatically bridge the semantic gap between what applications do and what users expect them to do from their description. Our study, however, focuses on reviews written by users, which do not always follow rigid grammatical structures [12, 13, 14].

Recent works by Gruber et al. [15] also focused on mining privacy policy of apps to identify critical discrepancies between developer-described app behavior and permission usage. Further, Sadeh et al. [16] focus is on comparing the practices described in privacy policies to the practices performed by smartphone apps covered by those policies. Although these two works are comparing privacy policy of the apps, it is not based on real experience, and it is based on the documents provided by the developers.

Bugiel et al. [17] measured the impact of user reviews on Android app security and privacy. This method first measures the security and privacy relevant reviews (SPR), and then for each app version mentioned in the SPR, they use static code analysis to extract permission-protected features mentioned in the reviews. However, their study does not show if the mentioned privacy and security leaks is actually leaking in the application or not.

2.3 App Security Evolution

Calciati et al. [18] studied how the permissions requested by apps evolve across different app versions. Their results show that many newly requested permissions are in apps evolution. Felt et al. [19] identified the violation of least-privilege by app developers, which is unfortunately a long-standing problem. Past research has also investigated how users should be confronted with permission requests, most noticeably early studies by Felt et al. [20, 21]. More disruptive proposals try to eliminate the explicit role of the user for permission granting as proposed by Roesner et al. [5] or the use of machine learning as proposed by Wijesekera et al. [4] and Olejnik et al. [3]. Most recently, different works pointed out the risks of third party libraries, in particular of advertisement libraries [22, 23, 24, 25] and other vulnerable libraries [26, 27]. However, to the best of our knowledge, we are the first to study how much app details can be connected to the security and privacy related reviews.

3 Methodology

In this section, we discuss the methodologies that we used for our study in order to answer our research questions. To achieve this, we perform two types of activities: identify what the users say about the apps in the reviews, and identify the behaviour of the app. The following sub-sections describe this process in details. The Figure 1 depicts the process flow.

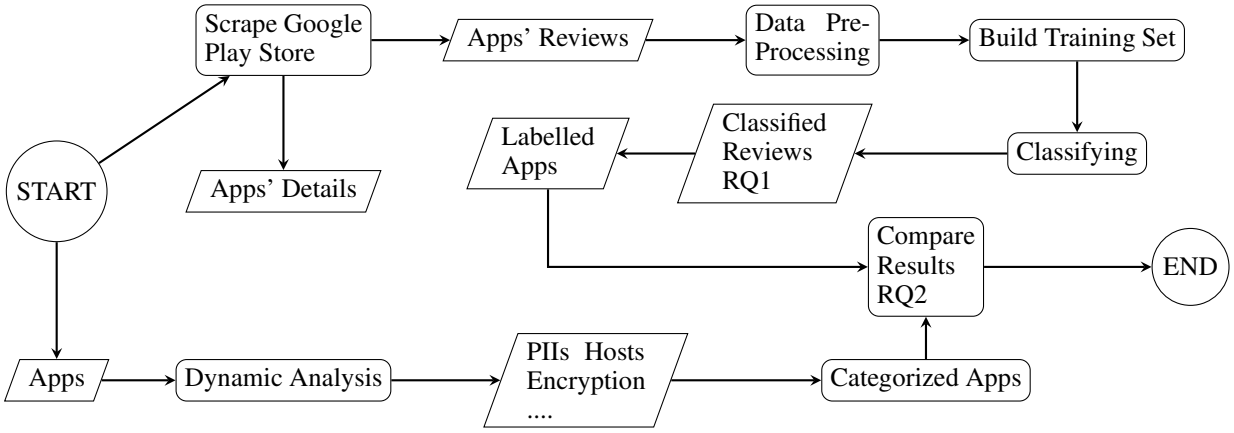


Figure 1: Process Flow Diagram

3.1 Analysing Reviews

In this section, we discuss the technical details for scraping and classifying the reviews.

3.1.1 Google Play Scraper

We built a custom web scraper to collect Android applications' details and reviews from Google Play. As previous studies [28, 29] have shown that only a small fraction of free applications on Google Play accounts constitute the bulk of the application downloads, we collected the details of the top free apps that are the most popular in Google Play. This resulted in 539 distinct applications. We also used another scraper to mine reviews of these apps.

Table 1: List of Keywords

Keywords		
Privacy	Security	Safe
Secure	Permission	Identity
Personal	Virus	Malware
Malicious	Access	Fishy
Phishing	Fishing	Stealth
Steal	Thief	Creepy

We scraped reviews that were written in English language only. After we had scrapped all the reviews for the apps, we pre-processed the reviews. Since user reviews are often written on smart phones, they tend to be short and usually contain grammatical mistakes or typos [12, 13, 14]. For this sake, we did the following steps:

- Removed links, unwanted characters, non-ASCII characters, special characters, etc.
- Removed english stop words like “a”, “the”, “from”, “is”, etc.
- Lemmatized the text.

The output of the above text pre-processing resulted in the final data that we used for analysis.

3.1.2 Classifying reviews

In order to classify the reviews, we initially needed to build the training set. This is achieved by manually labeling the reviews. Our first step was to find reviews that can be potentially related to security and privacy. We decided to search into reviews using a list of keywords. We initially performed a literature review to identify some related keywords that users may use in their reviews when they describe any security or privacy related aspect. We searched based on these keywords and also manually examined some other reviews to identify other keywords. After some iterations, we could built a list of words that can be used to retrieve related reviews. Table 1 presents this list.

We counted the number of occurrences of the keywords in each review. We found that the maximum occurrences is 5 keywords in a single review. Eventually, for the training set, we picked all the reviews with 2 or more keywords (totally 2122 reviews). We also added 2000 reviews, randomly selected from the ones with 1 keyword. Finally, we added 1878 reviews without any keyword to create a fair representative of all the reviews. So our final training set contained a total of 6000 reviews.

The next task was to manually annotate these reviews as either *Related* or *Non-Related*. For labeling, we considered a review as *Related* if the review matched any of the below criterion:

- Concerns about their personal information stolen, illegally accessed, or shared with third parties without permission.
- Concerns about their Password or User-name(identity) safety.
- Concerns about hidden background activities of the application.
- Concerns about unrelated taken permissions.
- Concerns about application security/privacy in general.

Three of our team members were assigned the task of manually coding these reviews. Each of them individually coded all the 6000 reviews as related or non-related. Once the coding was completed by all the members, the results were matched. For most of the reviews, there was unanimity amongst all the members. For the remaining mismatched labels, we individually discussed to agree on a single label. After completing this process, our training set contained 936 reviews, labelled as *Related* while the remaining 5064 were labelled as *Non-Related*. In order to balance the training set, we performed SMOTE, a well-known over-balancing technique in practice [30].

With our training set, we are now ready to classify the reviews. We needed to evaluate the efficiency of four classifiers, “Naive Bayes”, “K-Nearest Neighbour”, “Single-Layer averaged Perceptron”, and “Support Vector Machine (SVM)”, known for dealing with text data. To evaluate the efficiency, we used 10-fold cross validation, one of the highly recommended methods for validation [31]. We identified that SVM achieved the best results as compared to the other classifiers. It achieved high accuracy, precision and recall; so we selected SVM as the ideal classifier for our analysis.

3.2 Dynamic analysis

In this section, we use established dynamic analysis methods [32] to measure the actual behaviour of the apps as it relates to privacy. In particular, we use an instrumented version of the Android Nougat operating system, which we deployed on actual Nexus 5X phones. This instrumentation monitored all network traffic, including TLS-secured traffic. Our instrumentation can attribute specific network transmissions to the responsible application and records where on the Internet the data was sent. We search through this network traffic to find the presence of types of PII including location data and persistent identifiers.

We tested each of the apps by installing it, granting all runtime permissions, and then using a UI fuzzer to automatically interact with the app for a period of ten minutes. After this time the app is uninstalled and its network transmissions are saved for processing. This processing involves applying a suite of decoders, such as base64 and gzip, to reveal the raw data being transmitted. This also includes a number of deobfuscation methods based on ad-hoc obfuscation methods that we have seen third party libraries use in practice.

3.2.1 PII Types

In the network traffic generated by our experiment, we search for the presence of two types of PII: data used to geolocate the user and persistent identifiers for tracking. Location data is either the GPS coordinates, or the SSID and MAC address of the connected WiFi router, which is a well-known surrogate for location. For persistent identifiers, we divide them into two categories: resetable, which consists of the resetable advertising ID (AAID), and non-resetable, which consists of all other identifiers, including the android ID, IMEI, network MAC address, and serial number.

We separate these types of tracking identifiers because Google recommends developers only use the advertising ID and no other identifier for advertising purposes, and further recommends to avoid bridging resets of the advertising ID by linking it with other identifiers. As such, sending the AAID alone is reasonable when compared to combining it with other non-resetable trackers like the IMEI and the MAC address.

Based on this information, we categorized each app into one of the following categories:

- Good: If the app does not leak any PII type or only the AAID
- SingleTracker: If the app only leaks a single tracker PII type and no other PII types
- MultiTracker: If the app leaks multiple tracker types of PIIs and no other PII types
- SingleLocInfo: If the app leaks only a single Location Info type of PII and no other PII types
- MultiLocInfo: If the app leaks multiple Location Info types of PIIS and no other PII type
- AAID & Tracker: If the app leaks AAID and at least one type of Tracker PII
- AAID & LocInfo: If the app leaks AAID and at least one type of Location Info PII
- Tracker & LocInfo: If the app leaks both Tracker and Location Info types of PIIS but not AAID, and
- All: If the app leaks all three types of PIIS; i.e AAID, Location Info and Tracker

This list does not mean to order them in terms of invasiveness. That is, some may consider location worse than trackers and others feel the opposite. Nevertheless, there is an implicit order based on the subset relation, where sending the AAID is better for privacy than sending the AAID *and* location.

3.2.2 Domains contacted

Another metric we used to measure app privacy is the number of different domains to which PII was sent. That is, one app may include a single advertising SDK while another includes half a dozen so as to maximize revenue. This metric is not perfect, as a half dozen "good" SDKs may still be preferable to one invasive one. Nevertheless, the number of places on the Internet that are collecting PII from users devices does indicate how the app developer that includes these SDKs feels about user privacy.

In order to evaluate this, we categorized each app to one of the following groups:

- Level 0: 0–1 domains received PII
- Level 1: 2 domains received PII
- Level 2: 3–6 domains received PII
- Level 3: more than 6 domains received PII

3.2.3 PIIs leaked to each domain

The total number of hosts communicated with by an app may not always reveal the actual nature of invasion. For example, while some apps may be leaking the same PII to a large number of domains, there can be some apps which leak a large number of PIIs to a small number of domains. The impact of these two behaviours would naturally be different. So in addition to the total number of PIIs leaked and the total number of domains contacted, we also elicited the number of PIIs sent by the app to individual domains. We calculated the maximum number of PIIs sent by any app to a single domain.

3.2.4 Number of permissions asked by the app

We initially extracted the different types of permissions that an app can ask for and labelled each of them as either "Normal" or "Dangerous" based on the protection level set in [33]. Dangerorus permissions protect sensitive user data and sensors, like camera, location, and contact lists. As of Android Marshmallow, apps show a runtime dialog asking about the permission at the time it is first used. Once we have this data, we evaluated the total number of dangerous and normal permissions that the app needs. In order to judge if the app is encroaching into the users' privacy and security, we are only concerned about the dangerous permissions; also the users will be aware of the dangerous permissions only as the app would specifically request for those permissions. So, for analyzing if the users are concerned about their privacy and security, we consider only the dangerous permissions in our study.

3.3 Statistical approaches

In order to answer RQ2, we needed to calculate the correlation between different categories obtained from methods described above. As we have different types of data (Numerical and Categorical), we used the following statistical methods.

1. **Cramér's V**: A measure of association between two categorical variables. This measure assumes a symmetrical approach; i.e., the correlation between the two variables does not depend on the order of the variables.
2. **Theil's U**: Also known as "Uncertainty Coefficient", a measure of categorical association. This measure is used to calculate the correlation between 2 categorical variables when they assume an asymmetric approach.
3. **Correlation Ratio**: A measure to calculate the correlation between 2 variables that have mixed data types; i.e., one of the variables is categorical type and the other is of the numerical data type.
4. **Non-parametric Statistical Significance Test**: In order to ascertain if the correlation measures obtained using the above methods are meaningful, we used two well renowned non-parametric statistical significance tests, **"Mann-Whitney U"** and **"Kruskal-Wallis H"**. Hereafter, we denote these two test as "MWU" and "KWH" respectively. We performed the significance test using the data from dynamic analysis on the categories of the apps from reviews' classifier.

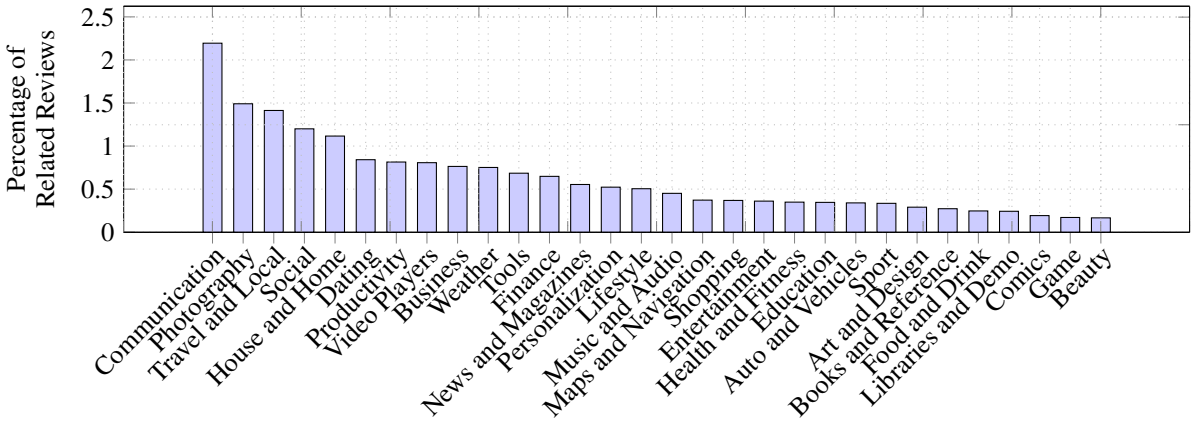
4 Dataset

To make a dataset of user reviews about mobile apps, we targeted the "top free" list of Google Play, the well-known Android app market powered by Google. The apps in this list are pretty popular and can potentially provide us more reviews to be assessed. This list typically contains 540 apps, but our dataset has 539 apps, as one of the apps was removed during the process of scraping. For each app, we scraped details such as category of the app, score, developer, title, number of reviews, number of installations, chosen by app store editors or not, description, content rating, and number of 1 to 5 stars ratings.

We also scraped the reviews of each app. Due to restrictions in Google API, we have access to the latest 4480 reviews of each app. This means that for less popular apps with less than 4480 reviews, we scraped all the reviews from the time it's been published. At the same time for some more famous apps, the limit of 4480 provide us reviews from last few months. For each review, we scraped details such as review text, review date, and current rating. For 539 top free apps in Google Play, we were able to scrape 2,186,093 reviews. By looking into the dataset, here we have some statistics:

1. App Reviews Rating: For each app, we have average of 2,090,749 ratings. Facebook, WhatsApp, Instagram, Messenger and Clash of Clans have more than 85M, 84M, 78M, 65M and 48M ratings respectively.
2. App Developers: These 539 apps are developed by 409 different developers, and 57 developers developed more than 1 app. Google with 29 developed apps, Voodoo with 13, Microsoft with 7, and Samsung and Amazon with 6 developed apps are top in the list.

Figure 2: Top categories based on the percentage of related reviews



3. App Category: Apps are from 32 different categories, and the Games, Entertainment, Tools, Social and Shopping are the categories with the highest number of apps having 217, 45, 28, 24 and 24 apps respectively.
4. App Installation: Around 60% of the apps have been installed more 10 million times.
5. App Content Rating: Less than 40% of the apps have content rating restrictions. Out of these, the majority have limited the audience to "teens".
6. Chosen by editor: 106 apps are chosen by editor of the Google Play.

5 Results

In this section we present the results from our experiments to evaluate our research questions.

5.1 Research Question 1

Regarding experiment configuration mentioned in Section 3.1.2, the classifier labelled 10,972 reviews as the related ones to security and privacy concerns. This means in only around 0.5% of all the reviews in our dataset, users write to mention a security/privacy concern. In order to provide more insights about the results, we have looked into the related reviews from different viewpoints:

5.1.1 Apps' categories

Figure 2 shows the percentage of related reviews per category. The top 30 categories are sorted and shown in this figure.

5.1.2 Reviews' Ratings

In our experiment we have classified the related reviews to security and privacy concerns, but we have not determined if the user is complaining of an app's functionality regarding his concerns or is praising it. To provide a good insight, an efficient way can be leveraging the rating of reviews. We consider reviews with 4 and 5 stars as *"positive"* reviews, reviews with 1 and 2 stars as *"negative"* ones, and reviews with 3 stars as the *"neutral"* reviews. Following this categorization, around 57% of reviews tagged as negative, 9% as neutral and 34% as positive. Shown in Figure 3, we have included the total number of reviews per rating in blue bars and the percentage of related reviews per rating stars with red bars.

5.1.3 Apps' Developers

Another interesting perspective to notice is the role of developer in changing users' thoughts. To assess this, Figure 4 shows the top 20 developers with highest number of apps along with the percentage of the negative related reviews for each.

Figure 3: Number of all the reviews and percentage of related reviews per rating stars

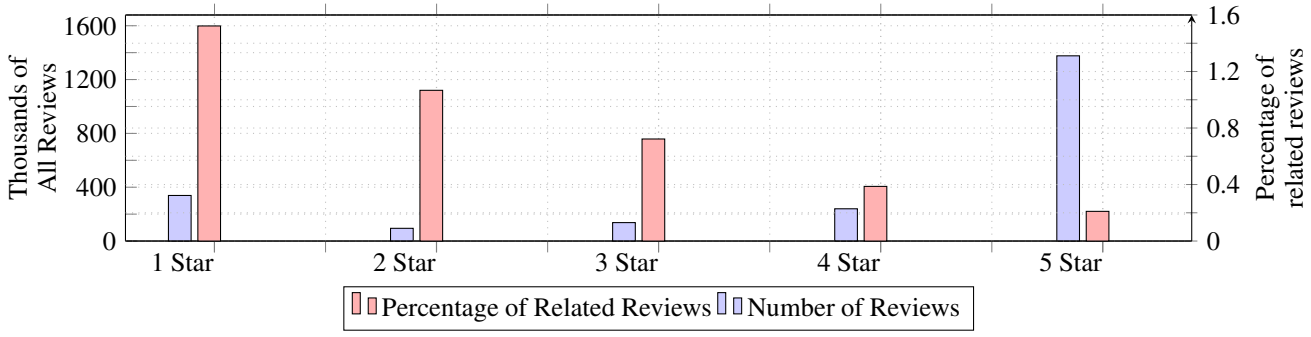
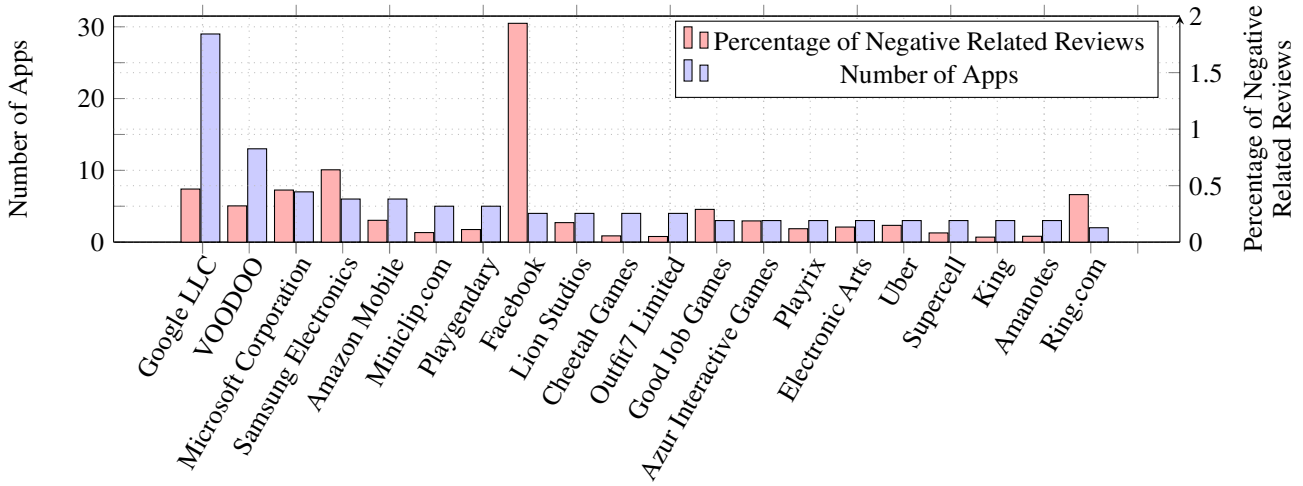


Figure 4: Top 20 developers based on number of apps along with the percentage of negative related reviews



5.2 Research Question 2

To answer the second research question, we needed to assess the correlation between the users' reviews and the functionality of the apps, determined from the analysis mentioned in Section 3.2.

At the very first step, we need to categorize apps based on users' thoughts about them. To achieve this, we tag related reviews in the same way mentioned for RQ1 in 5.1.2. Then following the definitions mentioned in Table 2, we tag the corresponding apps.

Furthermore, from the dynamic analysis mentioned in Section 3.2, we found PII leaks over different hostnames for each app. We also found the number of *dangerous* permissions asked by the apps. We call a permission "dangerous" when it needs two step confirmation from the user; in other words, it will be shown to the user at runtime through a pop-up dialogue. In order to make a meaningful comparison, we start considering different perspectives for categorizing

Table 2: Definition of the tags for apps

Tag name	Definition	Number of apps
Good	The ratio of positive reviews to all related reviews is greater than the same ratio for negative ones by 20%.	9
Neutral	The difference between the ratio of positive and negative reviews to all related reviews in less than 20%.	9
Bad	The ratio of negative reviews to all related reviews is greater than the same ratio for positive ones by 20%.	38
Not Discussed	The ratio of related reviews to all reviews is less than 1%.	483

Table 3: Description of the perspectives used for analysis

Perspective	Statistics	Type of the data	Code
The number of hostnames contacted by an app (The chattiness the apps).	Average of 1.8 hostnames per app.	Numerical	CH
If an app is sending out the location information of the user or not.	51 apps are sending.	Categorical	LOC
If an app is sending out AAID along with another tracker, regardless of the hostname (Bridging AAID).	100 apps are sending.	Categorical	BA
If an app is sending out AAID along with another tracker, over a single hostname (Bridging AAID).	93 apps are sending.	Categorical	BAH
The maximum number of PIIs sent over a single hostname by an app.	Average of 0.7 PIIs per app.	Numerical	MPH
The number of important permissions asked from user.	Average of 3.6 permissions per app.	Numerical	PE

Table 4: Correlation achieved between app's categories based on reviews and apps' categories based on perspectives from functionality analysis

Perspective	CH	LOC			BA			BAH			MPH	PE
Approach	CR	CV	TU1	TU2	CV	TU1	TU2	CV	TU1	TU2	CR	CR
Apps categorization based of reviews	.080	.132	.018	.024	.025	.010	.009	.000	.008	.007	.055	.365
Apps categorization based of reviews*	.079	.087	.011	.012	.012	.010	.007	.000	.009	.007	.054	.356

apps using the analysis results. The description of perspectives along with some related statistics, the type of data, and their corresponding abbreviations (used for further comparison) are mentioned in Table 3.

As the categorization of the apps based on reviews gives us categorical data, for correlation calculation we have categorical vs categorical data and categorical vs numerical data. As mentioned in Section 3.3, for the categorical vs categorical data we have used the symmetry approach of "Cramer's V" (hereafter shown by CV) and asymmetry approach of "Theil's U" (hereafter shown by TU1 and TU2[1]). For numerical vs categorical data also "Correlation Ratio" approach has been used (hereafter shown by CR).

The Table 4 shows the result. The third row shows the correlation between apps' categories based on related reviews and categories of apps in different perspectives. Considering the Figure 3, we noticed that the negative reviews form the majority of the related reviews. From this, we can assume that people do not usually praise apps for security and privacy concerns; therefore the apps with tag of "Not discussed" are actually "Good". Hence we replaced the tag of "Not discussed" to "Good" (in Table 2). This provided us a new correlations whose results are listed in the last row.

We have also used non-parametric statistical significance tests to assess the relation between categories of the apps in terms of reviews and in terms of their actual behavior. The Table 5 provides the p-values for both the *MWU* and *KWH* tests (row 3). Following the same assumption as above, in the last row we have the p-values for these tests, where all the apps with tag of "Not discussed" have been replaced with the tag of "Good".

Table 5: P-values obtained from non-parametric statistical significance tests

Perspective	CH		LOC		BA		BAH		MPH		PE	
Statistical Test	KWH	MWU	KWH	MWU	KWH	MWU	KWH	MWU	KWH	MWU	KWH	MWU
Apps categorization based of reviews	.706	.359	.041	.022	.330	.170	.647	.331	.467	.238	.649	.331
Apps categorization based of reviews*	.080	.040	.795	.398	.941	.471	.767	.384	.120	.060	.000	.000

[1]TU1 shows the correlation between first set and second one and TU2 shows the vice versa.

6 Discussion

After running the classifier, we found around 0.5% of the all reviews are related to security and privacy concerns. We looked into the related reviews from different viewpoints. In terms of categories of the apps, although Game, Entertainment, Tools, Shopping, and Social are the categories with the highest reviews in our dataset, when it comes to the ratio of security/privacy related reviews, we see different categories on top of the list. It is not surprising to see categories such as Communication, Photography, Travel and Local, and Social are on top of the lists, as the users may be worried about the security of their personal information in communication and social mobile apps and also they may be concerned about the access of apps to some of their private information such as photos and location (cf. Figure 2). Considering the rating of the reviews in Figure 3, it seems most of the time people complain about their concerns, as the number of related reviews with 1 star is the highest. In terms of the role of developer, we expected to see that the developers with higher number of apps in our dataset, have higher number of reviews and consequently the same ratio of negative related reviews. Initially we noticed that the top 20 developers with most apps in our dataset have the most reviews as well (with the same ordering). Then after plotting the percentage of negative related reviews, we noticed for some developers like "Facebook" and "Samsung", ratio of negative related reviews to all reviews are higher than expected (Figure 4). This shows us that the developer of an app can potentially influence the users' perception.

After the process of dynamic analysis, we were able to match the actual behavior of the apps with users' judgment. The third row of the Table 4 shows the correlation between apps' categories based on related reviews and categories of apps from the different perspectives. In the first look, we noticed that excluding the perspective of the last column (which relates to the number of permissions asked from users), in all the others, almost there is no correlation. Furthermore based on the results we obtained from the first research question, we assumed that people do not usually praise apps for security and privacy concerns and we changed the apps with tag of "Not discussed" to "Good" for better understanding. This resulted in new correlations shown in the last row. Again, excluding the last column, there is almost no correlation between different perspectives and apps' categories based on users' reviews. Considering that the only correlation happens based on the number of permissions asked from user, we can say the main criterion of the users for judging the functionality of the apps in terms of privacy and security is the number of permissions asked. In another viewpoint, users do not have any other promising criterion to help them decide how to feel about an app. For the significance tests (cf. Table 5), we assumed the significance level (the probability of rejecting the null hypothesis when it is true, denoted by α) to be 0.05. We can see that the "p" value is very small for PE (row 4); so we can reject the null hypothesis and can confirm that the number of permissions asked is related to the categorization of the apps. Again for column LOC, the p value is smaller than α in row 3; hence, we can interpret that LOC also affects the app categorizations. This definitely makes sense because if an app sends out location details, it needs to ask for permission from the user in most of the cases. Therefore, the results under LOC are affected in the same way as PE. So, in these aspects, we can say that the users' reviews can be a signal of the app's behavior. On the other side, if we consider the p-values under CH, it is greater than α for the majority of the cases; hence we failed to reject the null hypothesis. We have the same results for BA, BAH and MPH perspectives. So in these cases, we can say that the users' review are not a good indicator for the behavior of the apps.

The main limit of our study was the restriction of Google API for accessing to reviews of an app. For some of the famous apps, we only had access to the latest reviews for last few months. The reviews in short time intervals can be biased and may not provide a fair representation of users' perception.

We found that the number of permissions can considerably influence users' thoughts. Therefore, a main step for future can be evaluating if the higher number of permissions necessarily results in worse functionality in terms of security and privacy. Along with this, investigating the most efficient practices for addressing privacy/security concerns while getting high number of permissions, is always an interesting topic. All these ideas can be equipped by approaches for making users to feel safe and comfortable while granting permissions. On the other hand, developing tools and guidelines to increase the number of users' criterion for judging their privacy violations can be a good step in the future.

7 Conclusion

In our study, we initially scraped details and reviews of 539 top free mobile applications of the Google Play Android app market. We made our dataset with the total number of 2,186,093 of reviews. After pre-processing phase, we ran our classifier, SVM, which provided us the best performance according to 10-fold cross validation. In our experiment, 0.5% of reviews are classified as related to security and privacy concerns. Our analysis showed that users are more concerned for the apps from categories such as communication, photography, travel, and social. They also mostly provide their security/privacy related reviews along with low ratings (1 and 2 stars). We also noticed that users may get influenced by the developer of an app. It means, they may trust or distrust a developer.

In order to assess the judgment of users, we performed dynamic analysis to see the actual behavior of the apps. We found a correlation between the judgment of users for an app and the number of permissions asked from user by it. We also noticed some correlation between the categories of apps based on user reviews to those apps that send out location information. In all other perspectives such as "number of domains contacted" and "bridging AAID", no significant correlation was observed. This can be originated from the fact that permissions are the main criterion for judging the amount of access of the apps to personal data.

References

[1] Zhengyang Qu, Vaibhav Rastogi, Xinyi Zhang, Yan Chen, Tiantian Zhu, and Zhong Chen. Autocog: Measuring the description-to-permission fidelity in android applications. In *Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security*, pages 1354–1365, 2014.

[2] Alessandra Gorla, Ilaria Tavecchia, Florian Gross, and Andreas Zeller. Checking app behavior against app descriptions. In *Proceedings of the 36th International Conference on Software Engineering*, pages 1025–1035, 2014.

[3] Katarzyna Olejnik, Italo Dacosta, Joana Soares Machado, Kévin Huguenin, Mohammad Emtiyaz Khan, and Jean-Pierre Hubaux. Smarper: Context-aware and automatic runtime-permissions for mobile devices. In *2017 IEEE Symposium on Security and Privacy (SP)*, pages 1058–1076. IEEE, 2017.

[4] Primal Wijesekera, Arjun Baokar, Lynn Tsai, Joel Reardon, Serge Egelman, David Wagner, and Konstantin Beznosov. The feasibility of dynamically granted permissions: Aligning mobile privacy with user preferences. In *2017 IEEE Symposium on Security and Privacy (SP)*, pages 1077–1093. IEEE, 2017.

[5] Franziska Roesner, Tadayoshi Kohno, Alexander Moshchuk, Bryan Parno, Helen J Wang, and Crispin Cowan. User-driven access control: Rethinking permission granting in modern operating systems. In *2012 IEEE Symposium on Security and Privacy*, pages 224–238. IEEE, 2012.

[6] Dennis Pagano and Walid Maalej. User feedback in the appstore: An empirical study. In *2013 21st IEEE international requirements engineering conference (RE)*, pages 125–134. IEEE, 2013.

[7] Ning Chen, Jialiu Lin, Steven CH Hoi, Xiaokui Xiao, and Boshen Zhang. Ar-miner: mining informative reviews for developers from mobile app marketplace. In *Proceedings of the 36th international conference on software engineering*, pages 767–778, 2014.

[8] Fabio Palomba, Pasquale Salza, Adelina Ciurumelea, Sebastiano Panichella, Harald Gall, Filomena Ferrucci, and Andrea De Lucia. Recommending and localizing change requests for mobile apps based on user reviews. In *2017 IEEE/ACM 39th International Conference on Software Engineering (ICSE)*, pages 106–117. IEEE, 2017.

[9] Bin Fu, Jialiu Lin, Lei Li, Christos Faloutsos, Jason Hong, and Norman 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*, pages 1276–1284, 2013.

[10] David M Blei, Andrew Y Ng, and Michael I Jordan. Latent dirichlet allocation. *Journal of machine Learning research*, 3(Jan):993–1022, 2003.

[11] Rahul Pandita, Xusheng Xiao, Wei Yang, William Enck, and Tao Xie. {WHYPER}: Towards automating risk assessment of mobile applications. In *Presented as part of the 22nd {USENIX} Security Symposium ({USENIX} Security 13)*, pages 527–542, 2013.

[12] Xiaodong Gu and Sunghun Kim. " what parts of your apps are loved by users?"(t). In *2015 30th IEEE/ACM International Conference on Automated Software Engineering (ASE)*, pages 760–770. IEEE, 2015.

[13] Mita K Dalal and Mukesh A Zaveri. Opinion mining from online user reviews using fuzzy linguistic hedges. *Applied computational intelligence and soft computing*, 2014, 2014.

[14] Adelina Ciurumelea, Andreas Schaufelbühl, Sebastiano Panichella, and Harald 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)*, pages 91–102. IEEE, 2017.

[15] Johannes Feichtner and Stefan Gruber. Understanding privacy awareness in android app descriptions using deep learning. In *10th ACM Conference on Data and Application Security and Privacy*, 2020.

[16] Peter Story, Sebastian Zimmeck, Abhilasha Ravichander, Daniel Smullen, Ziqi Wang, Joel Reidenberg, N Cameron Russell, and Norman Sadeh. Natural language processing for mobile app privacy compliance. In *AAAI Spring Symposium on Privacy-Enhancing Artificial Intelligence and Language Technologies*, 2019.

[17] Duc Cuong Nguyen, Erik Derr, Michael Backes, and Sven 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)*, pages 555–569. IEEE, 2019.

[18] Paolo Calciati and Alessandra Gorla. How do apps evolve in their permission requests? a preliminary study. In *2017 IEEE/ACM 14th International Conference on Mining Software Repositories (MSR)*, pages 37–41. IEEE, 2017.

[19] Adrienne Porter Felt, Erika Chin, Steve Hanna, Dawn Song, and David Wagner. Android permissions demystified. In *Proceedings of the 18th ACM conference on Computer and communications security*, pages 627–638, 2011.

[20] Adrienne Porter Felt, Serge Egelman, Matthew Finifter, Devdatta Akhawe, David A Wagner, et al. How to ask for permission. *HotSec*, 12:7–7, 2012.

[21] Adrienne Porter Felt, Serge Egelman, and David Wagner. I've got 99 problems, but vibration ain't one: a survey of smartphone users' concerns. In *Proceedings of the second ACM workshop on Security and privacy in smartphones and mobile devices*, pages 33–44, 2012.

[22] Michael C Grace, Wu Zhou, Xuxian Jiang, and Ahmad-Reza Sadeghi. Unsafe exposure analysis of mobile in-app advertisements. In *Proceedings of the fifth ACM conference on Security and Privacy in Wireless and Mobile Networks*, pages 101–112, 2012.

[23] Ryan Stevens, Clint Gibler, Jon Crussell, Jeremy Erickson, and Hao Chen. Investigating user privacy in android ad libraries. In *Workshop on Mobile Security Technologies (MoST)*, volume 10. Citeseer, 2012.

[24] Soteris Demetriou, Whitney Merrill, Wei Yang, Aston Zhang, and Carl A Gunter. Free for all! assessing user data exposure to advertising libraries on android. In *NDSS*, 2016.

[25] Sooel Son, Daehyeok Kim, and Vitaly Shmatikov. What mobile ads know about mobile users. In *NDSS*, 2016.

[26] Sebastian Poeplau, Yanick Fratantonio, Antonio Bianchi, Christopher Kruegel, and Giovanni Vigna. Execute this! analyzing unsafe and malicious dynamic code loading in android applications. In *NDSS*, volume 14, pages 23–26, 2014.

[27] Michael Backes, Sven Bugiel, and Erik Derr. Reliable third-party library detection in android and its security applications. In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, pages 356–367, 2016.

[28] Nicolas Viennot, Edward Garcia, and Jason Nieh. A measurement study of google play. In *The 2014 ACM international conference on Measurement and modeling of computer systems*, pages 221–233, 2014.

[29] Nan Zhong and Florian Michahelles. Where should you focus: Long tail or superstar? an analysis of app adoption on the android market. In *SIGGRAPH Asia 2012 Symposium on Apps*, pages 1–1, 2012.

[30] Nitesh V Chawla, Kevin W Bowyer, Lawrence O Hall, and W Philip Kegelmeyer. Smote: synthetic minority over-sampling technique. *Journal of artificial intelligence research*, 16:321–357, 2002.

[31] Ron Kohavi et al. A study of cross-validation and bootstrap for accuracy estimation and model selection. In *Ijcai*, volume 14, pages 1137–1145. Montreal, Canada, 1995.

[32] Irwin Reyes, Primal Wijesekera, Joel Reardon, Amit Elazari Bar On, Abbas Razaghpanah, Narseo Vallina-Rodriguez, and Serge Egelman. "won't somebody think of the children?" examining coppa compliance at scale. *Proceedings on Privacy Enhancing Technologies*, 2018(3):63–83, 2018.

[33] Android Developers. `https://developer.android.com`, 2020. [Online; accessed 10-February-2020].