



Article

CIAA-RepDroid: A Fine-Grained and Probabilistic Reputation Scheme for Android Apps Based on Sentiment Analysis of Reviews

Franklin Tchakounté ^{1,*}, Athanase Esdras Yera Pagor ^{1,*}, Jean Claude Kamgang ²
and Marcellin Atemkeng ³

¹ Department of Mathematics and Computer Science, Faculty of Science, University of Ngaoundéré, Ngaoundéré A0C 0A9, Cameroon

² Department of Mathematics and Computer Science, National School of Agro-Industrial Science, University of Ngaoundéré, Ngaoundéré A0C 0A9, Cameroon; jckamgang@gmail.com

³ Department of Mathematics, Rhodes University, Grahamstown 6139, South Africa; M.Atemkeng@ru.ac.za

* Correspondence: tchafros@gmail.com (F.T.); athanase9@gmail.com (A.E.Y.P.)

Received: 31 July 2020; Accepted: 17 August 2020; Published: 27 August 2020



Abstract: To keep its business reliable, Google is concerned to ensure the quality of apps on the store. One crucial aspect concerning quality is security. Security is achieved through Google Play protect and anti-malware solutions. However, they are not totally efficient since they rely on application features and application execution threads. Google provides additional elements to enable consumers to collectively evaluate applications providing their experiences via reviews or showing their satisfaction through rating. The latter is more informal and hides details of rating whereas the former is textually expressive but requires further processing to understand opinions behind it. Literature lacks approaches which mine reviews through sentiment analysis to extract useful information to improve the security aspects of provided applications. This work goes in this direction and in a fine-grained way, investigates in terms of confidentiality, integrity, availability, and authentication (CIAA). While assuming that reviews are reliable and not fake, the proposed approach determines review polarities based on CIAA-related keywords. We rely on the popular classifier Naive Bayes to classify reviews into positive, negative, and neutral sentiment. We then provide an aggregation model to fusion different polarities to obtain application global and CIAA reputations. Quantitative experiments have been conducted on 13 applications including e-banking, live messaging and anti-malware apps with a total of 1050 security-related reviews and 7,835,322 functionality-related reviews. Results show that 23% of applications (03 apps) have a reputation greater than 0.5 with an accent on integrity, authentication, and availability, while the remaining 77% has a polarity under 0.5. Developers should make a lot of effort in security while developing codes and that more efforts should be made to improve confidentiality reputation. Results also show that applications with good functionality-related reputation generally offer a bad security-related reputation. This situation means that even if the number of security reviews is low, it does not mean that the security aspect is not a consumer preoccupation. Unlike, developers put much more time to test whether applications work without errors even if they include possible security vulnerabilities. A quantitative comparison against well-known rating systems reveals the effectiveness and robustness of CIAA-RepDroid to repute apps in terms of security. CIAA-RepDroid can be associated with existing rating solutions to recommend developers exact CIAA aspects to improve within source codes.

Keywords: reputation; android; application; sentiment analysis; reviews; security service; NLP; Google Play; polarity

1. Introduction

Due to its open-source nature, Android is the most popular mobile operating system [1] and it is expected to keep this popularity until 2023 [2]. Android markets, such as Google Play [3] and other third-party markets (AppChina [4], Anzhi [5]) play an important role in the popularity of Android applications. Their number has significantly increased from one million in 2013 to about 2,900,000 applications [6] making Google Play a rich place where people can satisfy their needs.

In order to maintain its business notoriety, Google emphasizes on the quality control of the applications in Google Play. This control aims to ensure aspects such as ergonomics, correct running of the application's functionality and security. The last aspect is a major challenge since it preserves confidentiality, integrity and availability of user data. On Android, most of the approaches use static analysis relying on static features related to applications [7–11], dynamic analysis relying on features observed during their execution [12–14] and hybrid analysis combining the both [15–17]. Despite Google Play Protect provided by Google, to filter threats, there are still malicious applications carefully designed by bad people to have an impact on the security and privacy of users [8,18]. The aforementioned solutions therefore need additional knowledge.

The better quality the apps are, the more satisfied customers will be. So satisfaction of Android customers makes a lot of money to Google. However, as noted by Woods and Moore [19], the market for information security products such as Android apps, is subject to asymmetry meaning that consumers lack information about quality of apps they acquire. As a result of that, they can easily be victims of attacks because they have consumed (unknowingly) low quality products called lemons. In the market ecosystem, three general approaches have been so far designed to address this issue. Liability laws charge vendors costs in the case of an ineffective product. However, it is not easy to prove that it is the vendor who defects the product. Another approach is to involve expert people to certify that a product is secure. This approach is constrained by the fact that vendor creates fake expertise and that security flaws is detected in real execution. The reputation is the last approach to evaluate effectiveness of products [20]. In this approach, marketing activities massively communicate about functionality and even they present the contrary of what products reflect.

Through Google Play, consumers are able to download and install products related to their needs. After exploitation, they experience app functionality and they can provide reviews and ratings helpful to prevent future consumers to take bad products. By doing that, Google is informed about claims and developer product vulnerabilities, and therefore takes measures to sanitize its market. These elements can therefore be useful to overcome issues of "lemons".

Evaluation by ratings in Google Play consists of giving a number in the scale [1,2,6–8]. The less it is, the less the user has been satisfied. Google assigns a sort of global reputation based on this rating by aggregating individual ratings. Evaluation by reviews is the way users can post textual reviews about usage experiences. Rating is subjective to the user because it is not possible to have the reasons for which this rate is provided. Reviews are much more expressive and are provided in text form by the user to appreciate, depreciate and recommend an application. Consumers also use these reviews to draw attention to shortcomings in terms of ergonomics, operations and security. Researchers have shown that there is often a big difference between rating and the reviews given by users [21]. In addition, these reviews can reveal several different feelings such as positive, negative and neutral on different aspects of the application [22]. Such reviews are useful and relevant to developers and application store owners to better understand their customers and to recommend improvements [23,24]. However, they are provided subjectively and descriptively, therefore harder exploitable for decision making whether or not to install the application.

As for any social ecosystem, an application (seen as an agent) should be associated with a reputation score to predict bad behaviors [25]. Returning to Android, reputation can be evaluated based on permissions requests [26–28], Application Programming Interface (API) calls [8], information flow analysis [29,30] and other features [9]. These approaches require prior installation and are based on app features. Unlike we would like to prevent installation through analysis of reviews. Interesting outcomes

are obtained, revealing that reviews are useful to understand user sentiments in regard to Android products with the use of machine learning and deep learning techniques [31–33]. This understanding comes from efficient extraction and summarizing of key topics from reviews [24,27,34–37], required to tell the developers exactly where to improve across updates [23,35,38–44]. The security aspect is a serious concern while updating. Concerning effectiveness of apps regarding the security point of view, literature provides with very few attempts. Mobile App Reviews Summarization (MARS) [45] and Security-Related Review Miner (SRR-Miner) [46] are two systems designed to summarize and mine security-related reviews to provide quick understanding to developers. Unlike evaluating security reputation, their aim is just to facilitate understanding and recommendations towards security. They are too coarse-grained since they deal with threat-related security keywords and the analysis is performed down to sentence-level. One with no knowledge about security would not be able to understand summaries and some knowledge is missing from the whole comment since only sentences with security-keywords are extracted. The last limitation is that authors do not consider negative sentiments, although the whole terms of comment are linked. Apart from SRR-Miner having a reliable dataset of keywords, the others are not consistent in selection of keywords. Tesfay et al. [47] evaluated security-related reputation based on user feedback, the number of users who installed the studied application and the reputation of vendors. This proposal is subjective, does not exploit sentiment analysis on feedbacks and is subject to false reputation in case there are few feedbacks. Their interest focuses more on functional aspects than specifically on security. Like traditional Google star rating subject to inconsistencies across user perceptions, SERS [48] is a scheme proposed to rate applications regarding security claims. Authors rely on two opposed ways; the static analysis of required permissions leaking data and the sentiment analysis of reviews. However, it is unclear how they proceed with keywords and polarities and they only deal with the confidentiality aspect. Authors are not specific to reviews concerned with the whole reviews of apps.

In light with all the aforementioned limitations, this work proposes, CIAA-RepDroid, a reputation scheme based on a review-level sentiment analysis of security-related reviews. CIAA-RepDroid exploits the probabilistic model Naive Bayes to estimate probability that the reputation of an app is likely good or bad. The specificity of the proposal is that we study more fine-grained aspects such as confidentiality, integrity, authentication and availability (CIAA) to provide enough ingredients to developers for improvement of future versions. More interestingly, with this system, a reputed app in term of security cannot be reputed in one of CIAA aspects. Experiments were conducted with 13 applications including live messaging, e-banking, social media and anti-malware including a total of 1050 security-related reviews and 7,835,322 functionality-related reviews. Results show that applications are badly reputed lacking effective measures for confidentiality, integrity, authentication and availability. Analysis of security-related claims show that confidentiality is the service the most neglected during the development of Android apps. Moreover, apps display good reputation in terms of functionality. This means that developers put a lot of efforts in functional aspect than security aspect. A quantitative comparison with SERS and Google rating shows that our approach is more precise and robust to estimate trust in terms of security. CIAA-RepDroid is able to recommend developers exact CIAA aspects to improve within their source codes. This work provides two main contributions.

- CIAA-RepDroid, a fine-grained security-related reputation based on sentiment analysis of security-related reviews and probabilistic classification model of polarities;
- a decomposition of reputation into confidentiality, integrity, authentication and availability reputations;
- experiments against existing rating system approaches demonstrating effectiveness of CIAA-RepDroid to related security flaws in Android apps.

The terms app and application are interchangeably used in the document. The rest of the paper is structured as follows. Section 2 provides a broad overview of authors who conducted similar works. Section 3 presents the proposed reputation model. Section 4 presents and discusses results obtained

with real applications and comparison against similar approaches. We conclude and provide further research to enhance our methods.

2. Literature Review

According to [9,24], the app source environment and user reviews contain relevant information about user experience and expectations. They recommend that developers and application store owners could leverage the information to better understand their customers. Their findings motivate this work which evaluates applications based on analysis of reviews to recommend developers and owners about security-related improvements. Any agent in social network ecosystem should be estimated a reputation score which is a predictor of future behavior based on previous interactions [10]. If they act positively in the past, they will be likely trust in the future because they are expected to act likewise. The reputation is a characteristic helpful to minimize dishonest agents. In this regards, every app in Android ecosystem is an agent which should be associated with a reputation score. This work predicts its actions based on analysis of experiences from users who installed. Based on review analysis, this work derives probabilistic model to assess reputation of any application and therefore estimates whether it is honest or dishonest seen as malicious or benign in terms of security aspects. Due to the unstructured aspects of texts, sentiment analysis comes therefore into play. Several proposals have come out to understand consumers's satisfaction through their comments and to evaluate app effectiveness in different aspects such as functionalities, security based on comments. This section presents six research trends related to our proposal.

2.1. Security of Android Apps

The first trend dedicated to security of Android is provided in three ways. Static analysis [7] aims at looking into app features such as permissions, source codes and other static characteristics to detect abnormal activities. Static analysis is not able to observe exploits deployed during runtime. This limitation is overcome in dynamic analysis [13], by scrutinizing data flows involved during runtime, which are helpful to identify malicious paths. Hybrid analysis [49,50] takes advantage of compromise between static and dynamic analysis and switch from one to another based on contexts. Unlike these works, we are about looking for reviews to evaluate and improve security-related health of apps on Google Play. Our work is complementary to the others. Reputation can also be evaluated through the risk induced based on permissions requests [26–28], API calls [8], information flow analysis [29,30] and complementary features [9]. The main flaw related to such reputation schemes is that they require the application to be installed. Even if they are successful, the application would have already caused damages. The purpose of our proposal is to prevent dangerous installations by investigating applications in the store.

2.2. Classification of Reviews Based on Polarities

The second trend includes mechanisms that look for best learning algorithms able to classify reviews based on sentiment polarities. Day and Lin [31] investigated the incidence brought by deep learning concerning sentiment analysis of Chinese reviews in Google Play. Long Short Term Memory (LSTM), Naive Bayes (NB) and support vector machine (SVM) have been used on a collection of 96,651 reviews on Google Play. Experimental results give an accuracy of deep learning of 94% which outperforms Naive Bayes (74.12%) and support vector machine (76.46%). Karim et al. [32,51] also study performance of machine learning algorithms while understanding reviews from Google Play. They found that Logistic Regression outperforms K-Nearest Neighbors, Naive Bayes and Random Forest in terms of precision, accuracy, recall and F1. Oyebode et al. [33] compared performance of five classifiers on 104 Android mental health apps by performing sentiment analysis of a total of 88,125 reviews. The best one comes out with F1-score of 89.42% and was therefore exploited to predict polarities helpful to study specific themes impacting satisfaction. Unlike these works, our aim is not investigate performance but security-related reputation of apps based on analysis of reviews.

2.3. Extraction and Summarization of Key Topics

The third trend is about extracting and summarizing key topics and features from reviews reliable to understand consumer preoccupations. Guzman and Maalej [34] proposed an approach which extracts key app features from reviews to help understanding quickly the opinions behind it. They obtain sentiments of the identified features using sentiment analysis and attribute to reviews general scores. They obtain via their method a precision of 0.59 and a recall of 0.51 on 7 apps. In the same vision, Noei et al. [24] developed an approach to extract the key topics from reviews that are more likely related to ratings of a specific category. Doing this will help the developer to concentrate on aspects which elevates user satisfaction. Experiments with a collection of 4,193,549 reviews of 623 Android apps in ten different categories show that there is a relation between app categories and review topics. Gu and Kim [37] designed Software User Review Miner (SUR-Miner), a tool to summarize and classify reviews. With this tool, one can extract essential aspects from reviews to assist developers to catch real improvements required by users. Khalid et al. [36] specifically study complaints in the reviews. They succeed summarizing 12 types of complaints relevant to users. This information is necessary for developers to consider proper user issues. Li et al. [27,35] developed mechanisms to understand the impact of an app's release by investigating the variation of user's sentiment in this app's reviews. Unlike to the above methods our topics concern security issues. We therefore extract key terms related to security complaints but to help identifying CIAA aspects from reviews. Then, we process by sentiment analysis coupled with Naive Bayes to determine a score referring to security-related reputation. Other works has been proposed to summarize comments on specific application aspects. These works can support our proposed method even if we specifically deal with security aspect.

2.4. Consideration of Reviews in the Updates

The fourth trend includes investigations of consideration of reviews in the updates. Nguyene et al. [38] established a connection between app reviews and the evolution of application's security and privacy to help developers cleaning apps. They investigated whether user reviews impact developers to consider their opinions for the next version. Experiments with a collection of 2583 apps with a total of 4.5 million user reviews showed that user really influence security evolution with their reviews across versions. However, they look for reviews completely dedicated to security and privacy. This is a limitation because within a review, users can talk about many aspects such as ergonomics and security simultaneous. Therefore knowledge is missed in their dataset. Unlike in our work where we look for security aspects in every review through CIAA keywords. Moreover, we avoid manipulating features related characterizing application such as permissions. Authors argue at the end, that future research on user reviews must be conducted to improve security. This recommendation motivates our work. ChangeAdvisor is a tool designed by Palomba et al. [39] to help developers in classifying reviews useful for app updates. ChangeAdvisor associates NLP, text analysis and sentiment analysis to automatically classify app reviews provided by consumers. WisCom is another tool proposed by Fu et al. [40] which analyzes reviews and ratings in mobile stores. Using regression and latent Dirichlet Allocation models, this tool successfully identifies inconsistencies in reviews and reasons justifying reject of apps by users. Authors of ChangeAdvisor and WisCom do not deal with defining reputation-based security aspects. There are more works dealing with mining reviews to improve next versions. Villarroel et al. proposed (Crowd Listener for releAse Planning) CLAP [41], a tool to categorize and cluster reviews by analyzing contents. Authors then assign priorities to clusters of reviews for upcoming releases. Gao et al. [42,43] developed a scheme to extract and categorize raised issues across reviews in different versions. Yu et al. [44] enhanced ChangeAdvisor providing ReviewSolver to locate the code part having problem. More research in this direction is provided in [23]. Unlike these works which are more general, ours aim is to provide fine-grained elements to help developers to improve security. Li et al. [35] proposed a scheme to identify the period during which

problematic updates have been performed although considerable negativity expressed in reviews. Our work does not deal with updates.

2.5. Mining Security Issues

The fifth trend deals with mining security issue reviews and user sentiments on that mined information. Hatamian et al. [45] proposed MARS, a tool which summarizes reviews and extracts privacy issues from them provided to help in decision making. Authors use machine learning, natural language processing and sentiment analysis to detect significant reviews related to privacy threats. Their method is able to provide precision, recall and F-score of 94.84%, 91.30% and 92.79%. This work considers the terms privacy and security as main keywords and determines the reputation of apps based on privacy threats such as tracking and spyware, phishing, unauthorized charges, unintended data disclosure, targeted ads, spam and others. We argue that (i) security and privacy as keywords are not fine-grained since one can express privacy or security issues with other words. (ii) Dealing with threats is also coarse-grained because it is hard for summarizing to semantically understand that a review poses privacy problem. (iii) For developers or users with no knowledge about privacy and security, recommendations are hardly understandable. To deal with these limitations, our objective is to find app reputation based on confidentiality, integrity, authentication and availability which are understandable basic terms in security. In other words, rather than to present in terms of threat, we present in terms of CIAA. For that, specific keywords indicating each security service are compiled. Tao et al. [46] developed SRR-Miner, a tool to summarize security problems and related sentiments of users. However their method has some problems in different aspects. The whole keyword can appear in a review. In this case, their categories will not be useful and therefore it is not possible to recommend exactly where improvements should hold. Moreover, categories are not precise since they are not related to security but to groups of applications. For example, the definition of the first category states that "... issues causing negative effect on devices..." but the negative effects are unclear. As the authors mentioned, SRR-Miner is not able to correctly label sentences with positive polarity including security services words such as authorization, authentication, login, etc. This fact reveals a limitation resulting from the fact to leave positive sentence and to compute polarity at the sentence level. In our work, we offer several aspects. We deal with positive, negative and neutral polarities at the review level. Categories on a fine-grained basis are security services including confidentiality, integrity, authentication and availability. A review poses a security issue if the application has some misbehavior in one of these services. The consideration of the three polarities coupled to CIAA categories allows the developer to easily circumscribe the problem. Moreover, their aim is to synthesize security-related comments by extracting security-related review sentences and identifying verbs representing the app's misbehavior from users' negative and neutral feedback. Our method's aim is to evaluate security-related reputation of the app based on sentiments extracted from comments.

2.6. Reputation and Rating Schemes

The sixth trend is about research proposing ways to evaluate application trust in some aspects. Tesfay et al. [47] put in place a private cloud to control user installations of Android applications and to keep track of feedbacks from users. They evaluate qualitative reputation of applications based on feedbacks, vendor's reputation and number of users who run the same application. This work holds for a private cloud, so customers are already known and their interests too. Their proposal does not propose formal scheme. It is subjective and somehow more related to permissions than feedbacks and is just informative. Our work deals with crowd users in Google Play whose interests are understood through reviews. Our proposal is formal based on sentiment analysis of comments provided by users in Google Play. In [48], authors provided a mechanism to rate apps based on static analysis of flows related to required permissions and sentiment analysis of reviews. Sentiments related to confidentiality are evaluated from reviews and internal abilities of the application to make possible data disclosure. It is unclear how they proceed with keywords and polarities. As result, they obtain

a rate of app meaning trust that the app does not disclose critical data. Unlike our scheme which considers four security aspects confidentiality, integrity, authentication and availability, their keywords only relate confidentiality. In our work, we are able to determine global security-related reputation and reputation for each of CIAA services. SERS appears to be similar to ours since it computes a value indicating the nature of apps. A global rating scheme is provided in popular app stores (such as Google Play and iOS store) to make customers evaluating based on their experience with an app. This per-user evaluation is expressed as number on a scale between one and five in the corresponding app and the system aggregates as the cumulative average of all individual user ratings over all the versions. The global rating value refers to satisfaction one could have after using an app or to a kind of reputation that users associate to an application. As reported in [24], different users with the same experience may provide different star ratings. Google rating is subject to inconsistencies across user perceptions. In our proposal, we take into account the current reviews to remove old reviews which may have been considered in previous versions. Moreover, our reputation relies on reviews.

3. Reputation Model

The methodology can be summarized in four main steps, illustrated in Figure 1. The first step concerns the collection of applications and selection of reviews related to security. The second step is to determine the polarity of the reviews based on the sentiment analysis. In the third step, review polarities are classified within security services such as confidentiality, integrity, authentication and availability. The fourth step is to determine the global reputation of an application based on the review polarities in terms of security-related keywords. Details of each step are provided below.

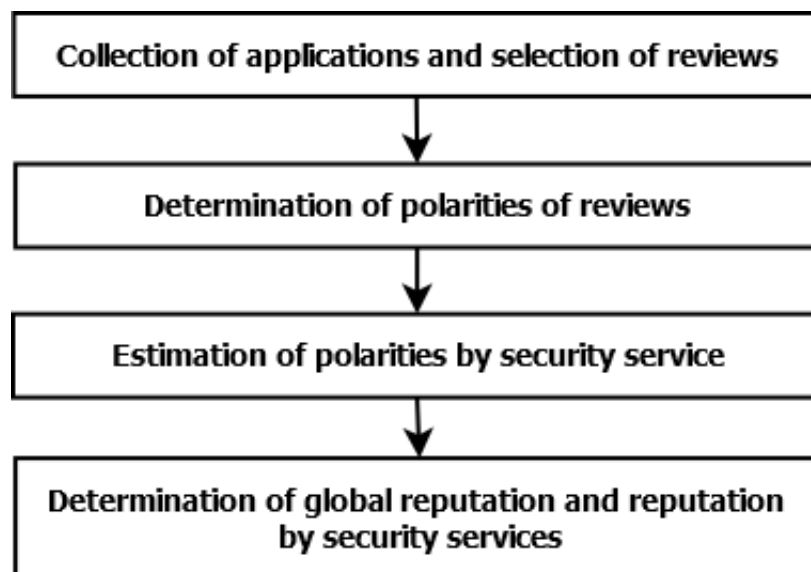


Figure 1. Research design.

3.1. Collecting Applications and Selecting Reviews

In this phase, we gather applications from Google Play based on reviews which point out security concerns and other criteria which can influence reputation of the application. These criteria include user ratings, number of users who downloaded the application and the number of reviews made by the customers. The process of extraction of applications and reviews is supported by the tool Google Play scraper [52] which is very flexible according to expected results and its ability to consider different languages. This tool also enables to filter reviews based on key terms, an important characteristic since we deal with security aspects. Reviews are filtered as follows. First, we exploit Appbot [53] to identify applications which show critical words from their reviews. Figure 2 shows an example of the app Instagram which has critical words such as Bug, Error, Crash. Appbot takes as input

sensitive applications that are expected to manipulate confidential information such as e-banking and messaging apps as well as app susceptible to prevent confidential data leaks such as antivirus. Once those applications are provided to Appbot, it gives for each app a list of critical words from the whole reviews of that application. We therefore select applications having reviews with critical words to perform deeper analyses. An analysis concerns extracting only reviews related to security aspects. Our method interchangeably translates keywords in French and English meaning that we consider both versions of the same keywords. The compilation of keywords is guided based on definitions of the main security services: confidentiality, integrity, availability and authentication. We refer to definitions of such services specifically in mobile environments [54]. Appendix A presents such keywords classified by security service with descriptions given considering the attacker side. For instance, concerning confidentiality the keyword “sniffing” is the art to passively observe information transmitted in the system. In the side of attacker, the one performing this action is not authorized to do that and tries to escalate privileges to capture information. Once we extract reviews of applications, some tools are used to mine keywords in reviews. The objective of mining is to confirm the effective presence of keywords in the reviews. We used two Search Engine Optimization (SEO) based tools, Alyze [55] and Keywordtool [56], to extract possible keywords from a review taken as input, associated with information such as number of occurrence and weight associated to each term of the review. They are of importance because it tells us whether our keyword appears or not. Even if it is associated with weight of 1, we consider the term as crucial to security aspect taken as a whole. Then we use Appbot, which is helpful since it deals with all the application’s reviews and refreshes automatically with added or deleted reviews. It provides the frequency of apparition of a keyword within application reviews. This information helps to confirm whether a keyword is relevant to consider. All the keywords have been found relevant to use in the proposed model. Figure 2 shows an example of the search.

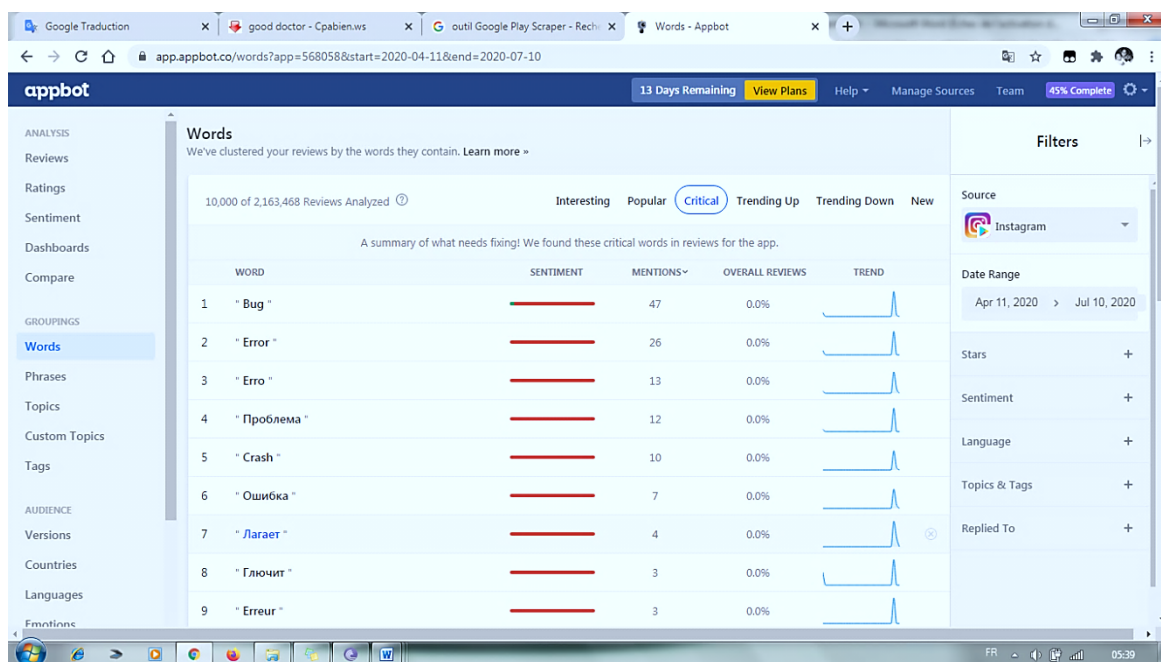


Figure 2. Excerpt of critical words of the Instagram application.

Appendix A depicts 10 reviews (in the original and translated forms) concerning security flaws made on the Instagram application. This example is exploited further on to illustrate the methodology steps.

3.2. Determination of Review Polarity

This step evaluates feelings that each user experiences when he makes a review. It is represented here as positive (+), negative (−) or neutral. This evaluation relies on Naive Bayes which is the most popular classification method for documents [57]. Its simplicity and flexibility makes it speedy and accurate to deal with large and diverse information manipulated in opinion mining. Within the scope of this work, a review is considered as a document. By definition, a Naive Bayes classifier is a classifier based on Bayes' theorem with the naive assumption that the entities are independent of each other. According to Bayes' theorem [58], for a characteristic vector $X = (X_1, X_2, \dots, X_n)$ and a class variable C_k , Bayes' theorem states that:

$$P(C_k | X) = \frac{P(X | C_k)P(C_k)}{P(X)}, \text{ For } k = 1, 2, \dots, K, \quad (1)$$

where, K is any choosing natural number and $P(C_k | X)$ is the posterior probability. $P(X | C_k)$ is the (previous) probability. $P(C_k)$ is the prior class probability (likelihood) and $P(X)$ is the previous probability of predictor (evidence). Using the chain rule, the probability $P(C_k | X)$ can be broken down as:

$$P(X | C_k) = P(x_1 \dots, x_n | C_k) = P(x_1 | x_2 \dots x_n, C_k)P(x_2 | x_3 \dots x_n, C_k) \dots P(x_{n-1} | x_n, C_k)P(x_n | C_k). \quad (2)$$

Two environments of analysis are used to obtain polarity: the sentiment analysis environment for Natural Language ToolKit (NLTK [59]) and the Spider environment [60]. These environments are well known for sentiment analysis and text classification. They evaluate whether texts express a positive, negative or neutral feeling. Using hierarchical classification, neutrality is determined then the polarity of feelings follows, but only if the text is not neutral. This classification is called Naive Bayes classification, which is nothing more than the application of Bayes rules for the formation of classification probabilities. This environment provides an overall feeling that a user has for a given application. For our work, it is a question of obtaining the polarity of feeling from a review by relying on the security aspects. We write Python script to exploit the keywords related to security services. Figure 3 depicts the polarity of each review from the script. In this specific case, the review has a negative polarity of 77.9% and a positive polarity of 22.1%. This review is therefore classified as negative.

```
IPython 6.1.0 -- An enhanced Interactive
Python.

In [1]: runfile('D:/spyder-py3/Test.py',
wdir='D:/spyder-py3')
Positive: 0.22058823529411764
Negative: 0.7794117647058824
*****
*****
Label : Neg

In [2]:
```

Figure 3. Result of the analysis of a review.

3.3. Polarity Classification by Security Service

This step determines feelings behind each review based on polarity and classifies them by security service. Based on model in the previous section, review polarities for the Instagram application are obtained and presented in Table 1. The review C2 for example, has a negative feeling with a probability of 0.84.

Table 1. Polarity of reviews related to the security of the Instagram application.

| Reviews | Positive Polarity (%) | Negative Polarity (%) |
|---------|-----------------------|-----------------------|
| C1 | 0.22 | 0.78 |
| C2 | 0.16 | 0.84 |
| C3 | 0.27 | 0.73 |
| C4 | 0.31 | 0.69 |
| C5 | 0.32 | 0.68 |
| C6 | 0.51 | 0.49 |
| C7 | 0.36 | 0.64 |
| C8 | 0.33 | 0.67 |
| C9 | 0.22 | 0.78 |
| C10 | 0.64 | 0.36 |

Now we look for classifying polarities by security service. For that, for a given application, we evaluate the distribution of apparition of security-related keywords in each security service. Then we match the sentiment to the distribution in each review. Concerning the application Instagram for example, Table 2 shows the breakdown of keywords by security service for reviews. Sentiments from reviews are obtained based on the model in Section 3.2.

Table 2. Distribution of security-related keywords.

| | Sentiment | Confidentiality Related Words | Integrity Words | Authentication Related Words | Availability Related Words |
|-----|-----------|-------------------------------|-----------------|------------------------------|----------------------------|
| C1 | Negative | 2 | 0 | 0 | 1 |
| C2 | Negative | 0 | 1 | 0 | 2 |
| C3 | Negative | 0 | 1 | 0 | 1 |
| C4 | Negative | 0 | 0 | 0 | 1 |
| C5 | Negative | 0 | 0 | 1 | 1 |
| C6 | Positive | 0 | 1 | 0 | 1 |
| C7 | Negative | 1 | 1 | 0 | 1 |
| C8 | Negative | 0 | 0 | 1 | 2 |
| C9 | Negative | 1 | 1 | 0 | 2 |
| C10 | Positive | 0 | 1 | 0 | 2 |

Based on Table 2’s outputs,

- C1 has a negative polarity in terms of confidentiality and availability;
- C2 has a negative polarity in terms of integrity and availability;
- C3 has a negative polarity in terms of integrity and availability;
- C4 has a negative polarity in terms of availability;
- C5 has a positive polarity in terms of authentication and availability; etc.

3.4. Determination of Reputation by Security Service and Global Reputation

The aim in this stage is to determine the reputation of an application based on the number of reviews recorded in Google Play, as well as their respective polarities while taking into account the security service that each review releases. We obtain the probability that an application has a good or a bad reputation, by subsequently relying on a probabilistic model defined as in Equation (3) [25].

$$E(p | \alpha, \beta) = \frac{\alpha}{\alpha + \beta}, \text{ with } 0 \leq p \leq 1; \alpha, \beta > 0, \tag{3}$$

where

- $\alpha = \sum N_{pos} = \alpha_C + \alpha_I + \alpha_A + \alpha_D$ (total number of positive polarity);

- $\beta = \sum N_{pos} = \beta_C + \beta_I + \beta_A + \beta_D$ (total number of negative polarity)

and

- α_C (resp.) β_C is the positive (resp. negative) polarity value for the service confidentiality;
- α_I (resp.) β_I is the positive (resp. negative) polarity value for the service integrity;
- α_A (resp.) β_A is the positive (resp. negative) polarity value for the service authentication;
- α_D (resp.) β_D is the positive (resp. negative) polarity value for the service availability.

Such that

$$\sum(\alpha_i, \beta_i) = 1. \tag{4}$$

Table 3 summarizes different polarity values for each review in the Instagram application. These values are obtained as follows: the value obtained after implementing the Python script on a given review, the polarities obtained, namely positive and negative (Table 3) are represented here while respecting the distribution of word numbers linked to the various security services mentioned above (Table 2). Thus to obtain the values α_i and β_i for a given review, we recover the number of words of the review distributed by security service, then we bring out the exact value of polarity obtained in Table 3 for each security service having a positive number (i.e., $\alpha_i, \beta_i > 0$). The final polarity value is therefore obtained as a function of the total number of words in the review highlighting the security aspect and the number of words distributed by security service (Table 3).

Table 3. Determination of (α_i, β_i) .

| Reviews | Polarity Index (+/−) | $N_{pos}(\alpha)$ | | | | $N_{neg}(\beta)$ | | | |
|---------|----------------------|-------------------|------------|------------|------------|------------------|-----------|-----------|-----------|
| | | Con. | Int. | Aut. | Ava. | Con. | Int. | Aut. | Ava. |
| C1 | − | 0.15 | 0 | 0 | 0.07 | 0.52 | 0 | 0 | 0.26 |
| C2 | − | 0 | 0.05 | 0 | 0.11 | 0 | 0.28 | 0 | 0.56 |
| C3 | − | 0 | 0.13 | 0 | 0.14 | 0 | 0.36 | 0 | 0.37 |
| C4 | − | 0 | 0 | 0 | 0.31 | 0 | 0 | 0 | 0.69 |
| C5 | − | 0 | 0 | 0.15 | 0.17 | 0 | 0 | 0.33 | 0.35 |
| C6 | + | 0 | 0.25 | 0 | 0.26 | 0 | 0.24 | 0 | 0.25 |
| C7 | − | 0.12 | 0.12 | 0 | 0.12 | 0.21 | 0.21 | 0 | 0.22 |
| C8 | − | 0 | 0 | 0.11 | 0.22 | 0 | 0 | 0.22 | 0.45 |
| C9 | − | 0.06 | 0.05 | 0 | 0.11 | 0.20 | 0.19 | 0 | 0.39 |
| C10 | + | 0 | 0.21 | 0 | 0.43 | 0 | 0.12 | 0 | 0.24 |
| Total : | | α_C | α_I | α_A | α_D | β_C | β_I | β_A | β_D |

We can therefore determine the reputation likelihood by security service by exploiting the formula of the probabilistic model $E(p | \alpha, \beta)$ for each service. We have:

- Confidentiality: $E_C = \alpha_c / (\alpha_c + \beta_c) = 0.6$.
- Integrity: $E_I = \alpha_I / (\alpha_I + \beta_I) = 0.7$.
- Authentication: $E_A = \alpha_A / (\alpha_A + \beta_A) = 0.2$.
- Availability: $E_D = \alpha_D / (\alpha_D + \beta_D) = 0.4$.

With:

$$\alpha_C = \sum N_{pos}(Con_i); \alpha_I = \sum N_{pos}(Int_i); \alpha_A = \sum N_{pos}(Aut_i); \alpha_D = \sum N_{pos}(Ava_i) \tag{5}$$

$$\beta_C = \sum N_{neg}(Con_i); \beta_I = \sum N_{neg}(Int_i); \beta_A = \sum N_{neg}(Aut_i); \beta_D = \sum N_{neg}(Ava_i). \tag{6}$$

The global reputation of a given application is the average of security services’s reputations of the same application.

$$E(p | \alpha, \beta) = \text{Mean}(E_C, E_I, E_A, E_D) = 0.32. \tag{7}$$

The global reputation must be within the interval [0–1] and is interpreted as follows: for an application with a good reputation in terms of security, the value of E must be greater than the value 0.5. An application with E less than 0.5 has a bad reputation and an application for a value equal to 0.5 has a neutral reputation. For instance, the Instagram application does not have a good reputation regarding the security aspect since this application has a reputation below 0.5.

In a more specific context, as shown in Table 3, we can infer the reputation of the application in terms of security services, which suggests to developers to improve security aspects in the next version. Interpretation of security service reputations is the same as for the global reputation.

4. Results and Discussions

This section presents the main findings from applying the proposed model to real applications in the Google Play. This section includes four experimentations: the first refers to study the reputation of Android applications on the security aspect. The second refers to evaluate their effectiveness in terms of functionality aspect. The third is dedicated to study how far developers ensure security aspect compared to functional aspect. The fourth is dedicated to compare our approach with SERS [48] and Google Play rating. The last experiment is dedicated to measure impact of changing keywords to keywords in SRR-Miner [46].

4.1. Dataset

We gathered 13 applications from Google Play in different categories: live messaging, social media, mobile banking, productivity and security tools. The intuitive justification of the selection of these applications is as follows: due to covid-19, people communicate more via live messaging and social media avoiding physical contacts. They are more interested in e-banking to perform facilities (paying bills, money transactions) and for that they install security tools and other tools to protect infiltrations. Studying applications falling in these categories is of high interest. Table 4 provides the number of reviews of each application in terms of security and functionalities as well as the aggregated rating based on number of votes.

Table 4. Dataset characteristics.

| Application | Category | Security Related Reviews | Functionality Related Reviews | Rating |
|--------------------|----------------|--------------------------|-------------------------------|--------|
| Instagram | Social media | 100 | 2,288,402 | 4.5 |
| Facebook | Social media | 150 | 1,479,779 | 4.2 |
| CIC | Finance | 90 | 1715 | 4.7 |
| Messenger | Communication | 150 | 101,065 | 4.2 |
| SGC Connect | Finance | 6 | 10 | 4.2 |
| Whatapp | Communication | 150 | 2,824,936 | 4.3 |
| QR scanner | Tools | 80 | 11,026 | 4.7 |
| Ecobank | Finance | 150 | 643 | 3.5 |
| Clean Master | Security tools | 98 | 139,070 | 4.5 |
| Security Master | Security tools | 103 | 78,840 | 2.8 |
| Age Scanner Finger | Entertainment | 105 | 118 | 2.3 |
| Express Union | Finance | 13 | 13 | 3.6 |
| BICEC Mobile | Finance | 2 | 5 | 3.8 |

4.2. Reputation Based on Security

Based on the reputation model, we obtain the results provided in Table 5.

Table 5. Reputation in terms of security; \sim : negative, +: positive.

| Application | Number Security Related Reviews | Application Polarity | $E(p\alpha, \beta)$ Security | $E(p\alpha, \beta)$ by Security Service | | | |
|--------------------|---------------------------------|----------------------|------------------------------|---|------|------|------|
| | | | | Con. | Int. | Aut. | Ava. |
| Instagram | 100 | – | 0.40 | 0.36 | 0.45 | 0.32 | 0.47 |
| Facebook | 150 | – | 0.31 | 0.42 | 0.36 | 0.11 | 0.33 |
| CIC | 90 | – | 0.23 | 0.12 | 0.34 | 0.18 | 0.26 |
| Messenger | 150 | + | 0.51 | 0.57 | 0.43 | 0.37 | 0.66 |
| SGC Connect | 6 | – | 0.18 | 0.13 | 0.25 | 0.15 | 0.17 |
| Whatapp | 150 | + | 0.58 | 0.38 | 0.58 | 0.68 | 0.67 |
| QR scanner | 80 | + | 0.51 | 0.46 | 0.57 | 0.34 | 0.68 |
| Ecobank | 150 | – | 0.30 | 0.52 | 0.33 | 0.11 | 0.22 |
| Clean Master | 98 | – | 0.39 | 0.43 | 0.38 | 0.19 | 0.43 |
| Security Master | 103 | – | 0.42 | 0.21 | 0.35 | 0.67 | 0.43 |
| Age Scanner Finger | 105 | – | 0.19 | 0.01 | 0.08 | 0.1 | 0.56 |
| Express Union | 13 | – | 0.41 | 0.43 | 0.33 | 0.64 | 0.23 |
| BICEC Mobile | 2 | – | 0.41 | 0.34 | 0.47 | 0.45 | 0.38 |

Table 5 presents the polarity of the applications on a defined number of reviews related to security services. Then, it presents the security related reputation of these applications in general and by security service. Some of them (QR scanner, Messenger, Whatapp) have almost average reputation. It appears that none of those applications has a very good reputation. In particular, it reveals for example that “Security Master” has a bad reputation (42%) in terms of security design. Users feel not comfortable while using this application. Developers have missed to put effort on the security aspect in general. However, authentication measures are desirable within the application (67%). The other services are not respected. Despite the fact that Whatsapp has an average reputation, its developers put efforts on integrity (58%), authentication (68%) and availability (67%) to guarantee non alteration of exchanges, to guarantee the service and to verify identity of communicators. The “Express Union” application aiming to provide banking operations has a bad reputation (41%) meaning that users feel not safe when using this application to perform banking transactions.

4.3. Reputation Based on Functionality

Table 6 shows the reputation of the same applications in terms of functional aspects. This is realized based on Appbot from which an evaluation of functionality on Android applications is provided depending on the point of view of ratings and review polarity.

Table 6. Reputation in terms of functionality.

| Applications | Number of Reviews | Positive Sentiments (%) | Negative Sentiments (%) | $E(p\alpha, \beta)$ Functionality |
|--------------------|-------------------|-------------------------|-------------------------|-----------------------------------|
| Instagram | 2,288,402 | 75% | 25% | 0.75 |
| Facebook | 1,479,779 | 63% | 37% | 0.63 |
| CIC | 1715 | 76% | 24% | 0.76 |
| Messenger | 1,010,765 | 65% | 35% | 0.65 |
| SGC Connect | 10 | 50% | 50% | 0.50 |
| Whatapp | 2,824,936 | 70% | 30% | 0.70 |
| QR scanner | 11,026 | 82% | 18% | 0.82 |
| Ecobank | 643 | 43% | 57% | 0.43 |
| Clean Master | 139,070 | 83% | 17% | 0.83 |
| Sécurité Master | 78,840 | 86% | 14% | 0.86 |
| Age Scanner Finger | 118 | 5% | 95% | 0.05 |
| Express Union | 13 | 15% | 85% | 0.15 |
| BICEC Mobile | 5 | 60% | 40% | 0.60 |

It reveals for example that “Security Master” has a very good reputation (86%) in terms of what it has been designed for. Users feel very comfortable while using this application. Unlike, Express Union which has a bad reputation (15%) meaning that users are not satisfied when using this application to perform banking transactions.

4.4. Reputation-Based Security vs. Reputation-Based Functionality

Figure 4 matches the reputation based on functionality with the reputation based on security. Apart from two applications Express Union and Age Scanner Finger, applications are always more

reputed in terms of functionality than in security. This fact can be explained by two situations. The first situation is that people feel more comfortable in terms of service than in terms of security. The second situation is that developers use vulnerable APIs which generate security flaws. Another remark is that developers of messaging applications tend to put more accents in security service than the others. This is the case of Messenger and Whatsapp. The correlation coefficient between reputation based security and reputation based functionality is 0.44 indicating that there is no relation between reputation based security and reputation based functionality. This result confirms that it is not possible to estimate security enforcement based on functionalities implementation. In other words, consumer reviews reveal that developers really lack to look into security aspects.

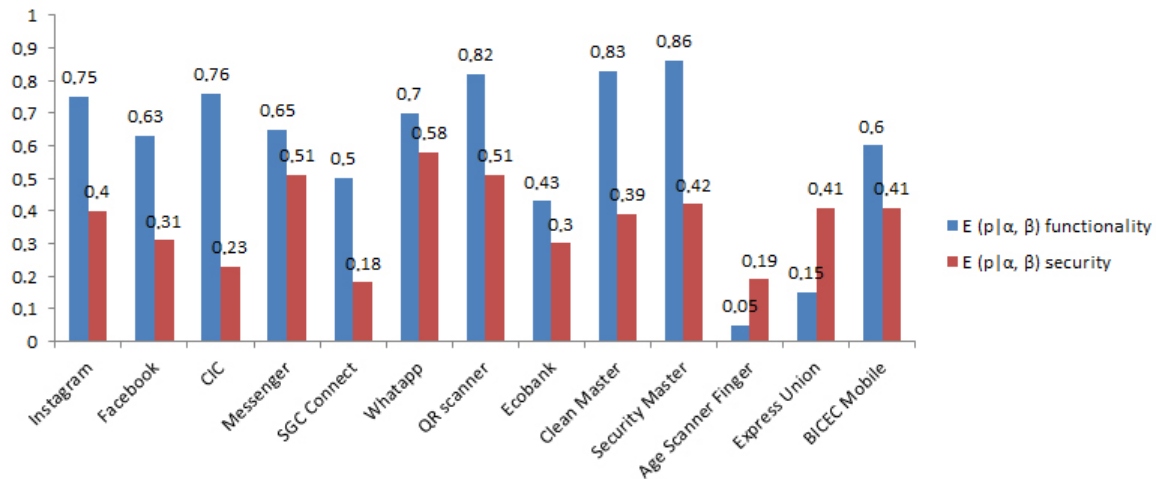


Figure 4. Reputation-based security vs. reputation-based functionality.

4.5. Comparison of CIAA-RepDroid to SERS and Google Play Rating

Table 7 summarizes results concerning reputation determined in our proposed scheme compared to ratings in SERS work and the traditional rating of Google. In this experimentation, the dataset of communication applications in [48] has been used. In this work, authors have computed their ratings shown in columns (R2), (R3) and (R4). They also have collected traditional rating available in column R5. What we did is to apply our approach to their samples to obtain R1. R11 is the fine-grained reputation in term of confidentiality, R12 in term of integrity, R13 in term of authentication and R14 in term of availability. As it is defined in the SERS’s work, R2 is a rating value of the application based only on sentiment analysis of the whole reviews (They consider all the reviews of the app to have R2); R3 is a rating value of the application based only on static analysis of required permissions; R4 is the global rating aggregating R2 and R3. To obtain R4. we have computed the average of weighted R2 and R3. The column #total reviews is the total number of reviews per application and # security reviews is the number of reviews related to security.

Table 7. Results about reputation and rating without gaps.

| App Characteristics | | Our Model | | | | SERS | | | Google | | | |
|---------------------|----------------|-------------------|------------------------|----------|----------|----------|----------|---------------------------|----------------------------|--------|--------------------|-------------------------|
| App Id | #Total Reviews | #Security Reviews | Global Reputation (R1) | Con. R11 | Int. R12 | Aut. R13 | Ava. R14 | Rating Based Reviews (R2) | Rating Based Analysis (R3) | Static | Global Rating (R4) | Traditional Rating (R5) |
| 1 | 2573 | 21 | 1.9 | 1.7 | 2.6 | 1.0 | 1.8 | 4.09 | 4.22 | | 3.66 | 4.1 |
| 2 | 2586 | 5 | 2.2 | 1.8 | 2.4 | 3.0 | 2.0 | 4.44 | 3.98 | | 3.95 | 4.6 |
| 3 | 2590 | 52 | 1.7 | 1.6 | 1.9 | 1.6 | 1.8 | 4.21 | 4.08 | | 3.82 | 4.5 |
| 4 | 2567 | 41 | 1.6 | 1.5 | 1.7 | 1.7 | 1.9 | 4.23 | 4.46 | | 3.86 | 4.5 |
| 5 | 2598 | 28 | 1.6 | 1.8 | 1.4 | 1.5 | 1.6 | 4.37 | 4.92 | | 4.46 | 4.5 |
| 6 | 2586 | 62 | 1.6 | 1.6 | 1.8 | 1.6 | 1.5 | 2.81 | 4.51 | | 3.43 | 4.3 |
| 7 | 2566 | 24 | 1.8 | 1.5 | 2.1 | 1.8 | 1.5 | 4.22 | 4.37 | | 3.95 | 4.1 |
| 8 | 2546 | 22 | 1.6 | 1.6 | 1.4 | 1.6 | 1.7 | 4.32 | 2.81 | | 4.52 | 4.4 |
| 9 | 2578 | 37 | 1.6 | 1.6 | 1.7 | 1.7 | 1.5 | 3.79 | 4.22 | | 2.51 | 4 |
| 10 | 2580 | 40 | 1.7 | 1.6 | 1.8 | 1.8 | 1.6 | 3.61 | 4.32 | | 4.25 | 4.1 |
| 11 | 2572 | 54 | 1.6 | 1.6 | 1.5 | 1.6 | 1.8 | 4.27 | 3.79 | | 4.19 | 4.6 |
| 12 | 2598 | 43 | 1.7 | 1.8 | 1.5 | 1.7 | 1.5 | 3.91 | 3.61 | | 3.44 | 4.6 |
| 13 | 2500 | 42 | 1.6 | 1.5 | 1.8 | 1.5 | 1.6 | 3.51 | 4.27 | | 3.29 | 4.2 |
| 14 | 2598 | 43 | 1.7 | 1.7 | 1.6 | 1.8 | 1.7 | 2.79 | 3.91 | | 3.89 | 4.2 |
| 15 | 2552 | 65 | 1.7 | 1.6 | 1.6 | 1.9 | 1.7 | 3.32 | 3.51 | | 2.55 | 4.1 |
| Average | 2573 | 39 | 1.70 | 1.63 | 1.78 | 1.72 | 1.68 | 3.85 | 4.06 | | 3.71 | 4 |

The first observation is that app ID #3 is the one having the higher number of reviews while app ID #15 is the one having more security-related reviews. Concerning column R5, we observe also that users have rated these apps of at least 4, meaning that they are very well satisfied about these apps. R2 slightly coincides with R4 except some apps such as 6, 9, 10, 13, 14 and 15 with which the gaps is around 1. This observation is similar between R3 and R4 and R4 and R5. The trend from the above observations (according to [24]) is that traditional ratings fit with rating based on reviews. Our work comes to explicitly determine security-related rating of app based on security-related reviews. Column R1 provides values of reputation less than 2.5 (in average 1.70), it means if we come to the original scale [0.1] that all the applications have a reputation under 0.5. Therefore, users all have negative sentiment while expressing their reviews. These negative sentiments are spread in all security aspects concerning CIAA for all apps where reputations are below 2.5 in average. Confidentiality is the aspect the most concerned with the smallest value of R1 in average. Developers should therefore reinforce this aspect. For instance, the app ID#2 with five (05) security-related reviews has a negative reputation of 2.2. This app is bad at confidentiality, availability but developers have inserted enough primitives for authentication (3.0/5). We also see that despite the fact that app ID#1 has a low reputation in security (confidentiality, authentication, availability), this app is almost good at integrity. This variability of reputation in security services strengthens of our model since it brings precise directions to developers in future improvements.

Now we compare the proposed reputation to the others. For that, we analyze absolute gaps between R1 and the others provided additionally in Table 8. We see that gaps are around 2 meaning that there is really a distance between satisfaction related to security, satisfaction provided by SERS and satisfaction traditionally expressed in Google. This observation is confirmed since the distance is almost constant in Gap R1-R2, Gap R1-R3, Gap R1-R4 and Gap R1-R5. Moreover, since there is a small number reviews concerning security among all reviews for each app, the security-related reputation should be distant from the one underlined by users (R5). However, these results are consistent because an application can have a good rate but a bad reputation in specific aspect. This is what is translated here. Ratings provided in Google show that users who rate are satisfied (4+ in R5) but in security (R1) they are not and also specifically in services CIAA. Here also our reputation finds its place. SERS aims at providing two ratings (i) based on security features of apps which overpass confidentiality and leak data (ii) based on reviews expressed by users. The second is biased since it deals with the whole reviews and therefore is not adapted to really security flaws to overcome compared to ours. Indeed, we have in average a rating of 3.70 in R2 showing that users are satisfied with apps, which is contrary to the objective of SERS to evaluate security rating. This lack in SERS is more visible since with five security reviews among 2586 (app ID#2), R2 provides positive satisfaction unless showing somehow that there are several dangerous vulnerabilities. This add-on is provided by our scheme through R11, R12, R13 and R14.

Table 8. Results about reputation and rating with gaps.

| App Id | Our Scheme | | SERS | | | Google | | Gaps | | | |
|---------|-----------------|--------------------|------------------------|---------------------------|-----------------------------------|--------------------|-------------------------|-----------|-----------|-----------|-----------|
| | # Total Reviews | # Security Reviews | Global Reputation (R1) | Rating Based Reviews (R2) | Rating Based Static Analysis (R3) | Global Rating (R4) | Traditional Rating (R5) | Gap R1-R2 | Gap R1-R3 | Gap R1-R4 | Gap R1-R5 |
| 1 | 2573 | 21 | 1.9 | 4.09 | 4.22 | 3.66 | 4.1 | 2.19 | 2.32 | 1.76 | 2.2 |
| 2 | 2586 | 5 | 2.2 | 4.44 | 3.98 | 3.95 | 4.6 | 2.24 | 1.78 | 1.75 | 2.4 |
| 3 | 2590 | 52 | 1.7 | 4.21 | 4.08 | 3.82 | 4.5 | 2.51 | 2.38 | 2.12 | 2.8 |
| 4 | 2567 | 41 | 1.6 | 4.23 | 4.46 | 3.86 | 4.5 | 2.63 | 2.86 | 2.26 | 2.9 |
| 5 | 2598 | 28 | 1.6 | 4.37 | 4.92 | 4.46 | 4.5 | 2.77 | 3.32 | 2.86 | 2.9 |
| 6 | 2586 | 62 | 1.6 | 2.81 | 4.51 | 3.43 | 4.3 | 1.21 | 2.91 | 1.83 | 2.7 |
| 7 | 2566 | 24 | 1.8 | 4.22 | 4.37 | 3.95 | 4.1 | 2.42 | 2.57 | 2.15 | 2.3 |
| 8 | 2546 | 22 | 1.6 | 4.32 | 2.81 | 4.52 | 4.4 | 2.72 | 1.21 | 2.92 | 2.8 |
| 9 | 2578 | 37 | 1.6 | 3.79 | 4.22 | 2.51 | 4 | 2.19 | 2.62 | 0.91 | 2.4 |
| 10 | 2580 | 40 | 1.7 | 3.61 | 4.32 | 4.25 | 4.1 | 1.91 | 2.62 | 2.55 | 2.4 |
| 11 | 2572 | 54 | 1.6 | 4.27 | 3.79 | 4.19 | 4.6 | 2.67 | 2.19 | 2.59 | 3 |
| 12 | 2598 | 43 | 1.7 | 3.91 | 3.61 | 3.44 | 4.6 | 2.21 | 1.91 | 1.74 | 2.9 |
| 13 | 2500 | 42 | 1.6 | 3.51 | 4.27 | 3.29 | 4.2 | 1.91 | 2.67 | 1.69 | 2.6 |
| 14 | 2598 | 43 | 1.7 | 2.79 | 3.91 | 3.89 | 4.2 | 1.09 | 2.21 | 2.19 | 2.5 |
| 15 | 2552 | 65 | 1.7 | 3.32 | 3.51 | 2.55 | 4.1 | 1.62 | 1.81 | 0.85 | 2.4 |
| Average | 2573 | 39 | 1.70 | 3.85 | 4.06 | 3.71 | 4 | 2.15 | 2.35 | 2.01 | 2.61 |

Now that we showed the completeness of what we propose against SERS reviews analysis, we come to compare to the second SERS rating. Since it relies on app features to detect data leaks (R3), we think that this rating is the more appropriate to compare with. Again, a distance exists between, R1 and R3. Unfortunately, the gap existing between R1 and R3 means that SERS’s second rating is not really expressive in terms of security. The average value in R3 is 4.06 which mean that applications do not leak any data by the use of risky permissions. This result is false since App ID #2 deals with a dangerous permission access (such as *READ_PHONE_STATE*). Our reputation expresses more reliably this situation.

From Table 9, we note some more important points concerning our proposal. The first is that app 2 has the highest reputation in terms of security and the same has the highest reputation in terms of functional aspect. This coincidence has a sense. In fact, it means that authors of this app made a lot of effort to strengthen it against security flaws. Indeed, only five security reviews are stated among 2586 reviews. This application happens to be considered as the more secure since few users confront security problems. The second is that the app 9 has the lowest reputation in terms of security and the lowest reputation in terms of functional aspect. This result simply means that users note through their 37 security reviews that this app has severe security pitfalls and users do not really recommend this app. The number of users put the Google rate to 4 since Google aggregates all the rates.

Table 9 shows the five reviews which relate security from our scheme. Effectively, they stipulate hacking, phishing via terms such as fake and denial of service through terms such as crash. Unlike SERS and Google that finally express functional aspects, we are able to see that developers miss to look at these aspects and customers suffer from consequences. Although our scheme reveals CIAA flaws, improvements should be made to indicate type of vulnerability or threat based on semantics and ontology in words.

Table 9. Security-related reviews of app ID #2.

| | Reviews |
|-------|---|
| app 2 | it’s always in my apps using my data, even after I uninstalled it and I am sick of it. been hacked |
| | Very very disappointing app don’t install this video app % fake |
| | it’s a after thought a copy cat FaceTime.... where is your imagination???? and what’s up with all the animation and Kitty shows this is supposed to be Facebook not fake book not babysitter book the mon where is the news we used to have the party that used to have the fun Facebook used to be now it’s just bogus people are raising their beers and their fingers to the whole it would have been better had it been passed off as a lesbian game show 0h.). yyyb be by |
| | always crashes and video never clear st |
| | Receiving calls causes crashing of the app. |

5. Study of Change from CIAA-RepDroid Keywords to SRR-Miner Keywords

Table 10 illustrates results about the determination of reputation with our CIAA keywords and the determination of reputation with SSR-Miner keywords published by the authors here. The objective is to investigate the impact of keywords to CIAA-RepDroid, which are key elements for sentiment analysis. Columns with identification of one (1) refer to processing with our approach. For example, # security Review 1 is the number of security-related reviews extracted using the CIAA-RepDroid keywords from the whole reviews whereas #security Review 2 is the number of security-related reviews extracted using the SSR-Miner keywords from the whole reviews. The three following points are to be discussed.

- The increase and the decrease of the number of reviews;
- the reduction of badness and goodness of app reputations;
- the stagnation of reputations despite the variation of the number of reviews.

Table 10. SRR-Miner keywords vs. our approach’s keywords. Conf.: confidentiality, Int.: Integrity, Auth.: Authentication, Avail.: Availability.

| App ID | # Total Reviews | # Security Review | | Global Reputation | | Conf. Reputation | | Int. Reputation | | Auth. Reputation | | Avail. Reputation | |
|---------|-----------------|-------------------|------|-------------------|------|------------------|------|-----------------|------|------------------|------|-------------------|------|
| | | 1 | 2 | 1 | 2 | 1 | 2 | 1 | 2 | 1 | 2 | 1 | 2 |
| 1 | 2573 | 21 | 26 | 1.9 | 1.8 | 1.7 | 1.7 | 2.6 | 2.6 | 1 | 1.5 | 1.8 | 1.7 |
| 2 | 2586 | 5 | 9 | 2.2 | 2.2 | 1.8 | 1.8 | 2.4 | 2.8 | 3 | 2.1 | 2 | 2.8 |
| 3 | 2590 | 52 | 41 | 1.7 | 1.7 | 1.6 | 1.6 | 1.9 | 1.6 | 1.6 | 1.7 | 1.8 | 2 |
| 4 | 2567 | 41 | 32 | 1.6 | 1.7 | 1.5 | 1.5 | 1.7 | 1.8 | 1.7 | 1.7 | 1.9 | 1.9 |
| 5 | 2598 | 28 | 43 | 1.6 | 1.6 | 1.8 | 1.7 | 1.4 | 1.5 | 1.5 | 1.8 | 1.6 | 1.3 |
| 6 | 2586 | 62 | 78 | 1.6 | 1.6 | 1.6 | 1.7 | 1.8 | 1.7 | 1.6 | 1.6 | 1.5 | 1.5 |
| 7 | 2566 | 24 | 35 | 1.8 | 1.7 | 1.5 | 1.7 | 2.1 | 1.8 | 1.8 | 1.5 | 1.5 | 1.7 |
| 8 | 2546 | 22 | 30 | 1.6 | 1.6 | 1.6 | 1.6 | 1.4 | 1.7 | 1.6 | 1.8 | 1.7 | 1.8 |
| 9 | 2578 | 37 | 29 | 1.6 | 1.7 | 1.6 | 1.7 | 1.7 | 1.8 | 1.7 | 1.6 | 1.5 | 1.7 |
| 10 | 2580 | 40 | 31 | 1.7 | 2.4 | 1.6 | 2.2 | 1.8 | 2.6 | 1.8 | 2.2 | 1.6 | 2.9 |
| 11 | 2572 | 54 | 68 | 1.6 | 1.7 | 1.6 | 1.8 | 1.5 | 1.7 | 1.6 | 1.5 | 1.8 | 2.1 |
| 12 | 2598 | 43 | 57 | 1.7 | 1.6 | 1.8 | 1.6 | 1.5 | 1.8 | 1.7 | 1.5 | 1.5 | 1.9 |
| 13 | 2500 | 42 | 34 | 1.6 | 1.7 | 1.5 | 1.5 | 1.8 | 1.8 | 1.5 | 1.7 | 1.6 | 1.7 |
| 14 | 2598 | 43 | 39 | 1.7 | 1.7 | 1.7 | 1.7 | 1.6 | 1.6 | 1.8 | 1.8 | 1.7 | 1.5 |
| 15 | 2552 | 65 | 81 | 1.7 | 1.6 | 1.6 | 1.6 | 1.6 | 1.6 | 1.9 | 1.8 | 1.7 | 1.6 |
| Average | 2572.7 | 38.6 | 42.2 | 1.70 | 1.75 | 1.63 | 1.69 | 1.78 | 1.89 | 1.72 | 1.72 | 1.68 | 1.87 |

Increase and decrease of the number of reviews. We observe fluctuations in the number of reviews from exploiting keywords of CIAA-RepDroid to exploitation of SSR-Miner keywords and vice-versa. In case of increase (from 1 to 2), the SSR-Miner keywords are able to catch more reviews than those in CIAA-RepDroid. In case of decrease (from 1 to 2), the CIAA-RepDroid keywords are able to catch more reviews than those in SSR-Miner. In one and another case, the terms to extract security-related reviews are complementary and can be associated.

Reduction of badness and goodness of app reputations. Concerning the global reputation, we note a slight variation of distance ± 0.1 apart from the application 10 which increases from 1.7 to 2.4 and apps where this reputation is constant. The first interpretation is that despite the difference between the set of keywords in both studies, the reputation obtained is quite the same. In other words, our reputation scheme gives the keywords the same semantics of representation security flaws. **CIAA-RepDroid** is therefore robust. The second interpretation concerning the decrease of badness or augmentation of goodness (in terms of security, from 1 to 2) is that several keywords do not appear in reviews in case 2 compared to case 1, in which it is possible to have several keywords in one review. Therefore, there are keywords unexploited in case 2. The third interpretation concerning the decrease of goodness or augmentation of badness (in terms of security, from 1 to 2) is that several keywords can appear in one of the reviews simultaneously in case 2 compared to case 1. Therefore, there are keywords unexploited in case 2. There are several cases where the two previous situations are amplified meaning the distance is much higher (For example: Application 1: authentication from 1 to 1.5; application 2: integrity from 2.4 to 2.8; authentication from 3 to 2.1. Application 3: integrity from 1.9 to 1.6). The justification remains the same but what we can add more is that the changeability of keywords can be helpful to signal risks in security services. Application 10 has a particular behavior since this CIAA-RepDroid considerably increases its global and specific reputations despite the reduction of reviews from 40 to 31 (global: +0.7, confidentiality: +0.6, integrity +0.6, authentication: 0.4. availability +1.3). After investigations, we discover that SSR-Miner keywords extracting the 31 reviews are also found in the same extracted CIAA-RepDroid reviews, but there are no keywords found within the 9 reviews left. Unlike in case 1 where several keywords were found in some reviews, keywords are evenly distributed, one per review. Something also interesting is that CIAA-RepDroid finds neutral polarities in neutrality in some statements. **CIAA-RepDroid is therefore adaptive to user expressiveness in the reviews.**

Stagnation of reputations. We note several applications where reputations are identical despite the variation of the number of reviews. For example, application 14 keeps the same reputations everywhere apart from availability. Despite the change of keywords, the reputation remains the same. Reputation is therefore independent of keywords. Our model reveals that there are semantically synonymous keywords and groups of keywords that refer to a keyword. A study of their vocabulary and semantics would be an asset to better understand this result.

Summary. Results in this section show CIAA-RepDroid keywords and SRR-Miner keywords can be complementary used. These results show effectiveness of CIAA-RepDroid to adapt to new keywords as well as robustness of CIAA-RepDroid to keep consistent security reputation.

5.1. CIAA-RepDroid Advantages

The proposed model contributes in some key points:

- CIAA-RepDroid reveals that reviews can be effective to indicate security flaws to overcome across updated versions;
- CIAA-RepDroid is able to prioritize security services in terms of reinforcement in the next version;
- CIAA-RepDroid is able to provide fine-grained security reputations. The model finds through opinions various CIAA security pitfalls to overcome before one installs inside the smartphone. It can be exploited to recommend developers and vendors.
- The reputation based on functionality is in major cases much higher than the reputation based on security except some applications;
- CIAA-RepDroid is more expressive and precise in terms of security than existing rating schemes;
- CIAA-RepDroid keeps a certain robustness despite the change of security-related keywords.

5.2. Limitations and Recommendations

Results show that the proposed model is effective in determining the security related reputation of an Android application. However, CIAA-RepDroid has some limitations:

- This model does not distinguish fake reviews from real ones. We assumed all reviews as true reviews.
- The consideration of the date of publication of reviews is not taken into account. Over time, applications can have improvements thanks to updates and customers change their feelings to positive.
- We have only considered one application source: Google Play Store. People can make reviews on the same application through different application stores.

This work firmly recommends app vendors to exploit fine-grained schemes such as CIAA-RepDroid to associate to the traditional rating to draw attention on security improvements about low-quality products.

6. Conclusions and Perspectives

Products are provided by app vendors to users through app stores. Despite offering rating schemes indicating user satisfaction, they lack efficiency to deal with low-quality products in specific aspects to improve their market. Security claims for instance should be overcome in this context. The objective of this research was therefore to propose an approach relying on review analysis able to recommend improvements to vendors and developers. In this regard, existing research (i) concentrates on ratings dealing with functional aspect to the detriment of security aspects, (ii) related to security are dedicated to summarization aspects and exploitation of code features for detection. This work proposed, CIAA-RepDroid, a fine-grained mechanism to evaluate security-related reputation based on probabilistic model and sentiment analysis. Additionally, we provided means to have reputations in terms of confidentiality, integrity, authentication and availability to fully identify security aspects where developers should put more efforts. We have experimented CIAA-RepDroid on 13 applications including a total of 1050 security-related reviews and 7,835,322 functionality-related reviews. Results revealed that in majority of cases, developers neglect security specifications in their development as well as security claims in their updates. Applications are more reputed in functional aspects than in security aspects. Moreover, confidentiality reputation remains the lowest, indicating that related measures are mitigated in their code. Therefore it is possible for attackers to access prohibited

data as the owner. This aspect must be investigated deeply. A comparison of CIAA-RepDroid to existing approaches demonstrated its efficiency, precision and completeness to present security flaws through reviews. A further investigation in the change of keywords related to another study showed that CIAA-RepDroid remains robust. Despite all these positive points coupled to its applicability, CIAA-RepDroid is subject to some pitfalls that require further investigations.

- CIAA-RepDroid will be extended to consider fake reviews;
- CIAA-RepDroid will include the fact reviews update with time and versions;
- CIAA-RepDroid will consider the fact that for an application, reviews are made in different stores.

Author Contributions: Conceptualization, F.T. and A.E.Y.P. ; methodology, F.T. and A.E.Y.P.; software, A.E.Y.P.; validation, F.T.; writing—original draft preparation, F.T., A.E.Y.P. and M.A.; writing—review and editing, J.C.K. and M.A.; supervision, F.T.; All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

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

Appendix A

Table A1. Keywords and descriptions.

| Security Aspect | Keywords | Justifications |
|-----------------|--|---|
| Confidentiality | Sniffing, spoofing | Sniffing refer to capturing and listening data that are transmitted across the whole system/network. The attacker violates confidentiality to achieve such actions. Spoofing helps the attacker to disguise as original entity to infiltrate a system. By doing that, one obtains privilege to access confidential information. |
| | Intrusion, usurpation, | An attacker needs to violate confidentiality by stealing identify to intrude the system. Then, he accesses unauthorized data. |
| | Steal, hacked, abuse | These terms refer to personal data used without the permission of the owner, and sometimes these data are used to harm the owner or other people. |
| | Malicious, virus, compromise, trojan | These terms refer to software that infects a computer system without the knowledge of its user by exploiting human or technical vulnerabilities They compromise confidentiality. |
| Authentication | Fake, alert, relentless | Some terms often used to deceive the user or influence opinion on a particular subject. The finality is to lure to bring out unknowingly confidential data. |
| | Scam, theft, breach, sniffing, spoofing, | We recall sniffing and spoofing since they rely on authentication to proceed. Additionally, attackers escalate authentication via scamming, identity theft and exploitation of vulnerabilities. |
| | Anomaly, authorized, erroneous, vulnerable, protect, certified | These terms refer to authentication in the sense that a user cannot access his account, but he nevertheless provides the correct information. The user can also spot suspicious activity in their account. During authentication process, a system must certify identities and prevent erroneous authorizations. Based on vulnerabilities, a malware profit to pose anomaly activities. |

Table A1. Cont.

| Security Aspect | Keywords | Justifications |
|-----------------|--|--|
| Integrity | Repudiation, nonsense, anomaly, abnormal | Once the state of the system changed and those data are altered, some related elements become nonsense. This alteration shows abnormal and anomaly behaviors throughout the system. Alterations are helpful to detect attacker traces. The latter generally uses repudiation measures to evade detections. |
| | Fraud, spam, lie, scammer, scam | These terms relate to social engineering attacks consisting of luring user to provide critical data. The reason why we put them into the integrity category is that their objective is the infiltration to the target. This infiltration makes the state changed. |
| Availability | Wrong, Error, Bug, Bugger, nonsense, Available, Crash, impractical | These terms refer to the inaccessibility or unavailability of a service or feature when using an application. Most of the time, users encounter access errors, applications that crash during their user, which sometimes leads to data loss. |

Table A2. Instagram reviews.

| | | |
|----------|------------|---|
| Review 1 | Original | il n y a aucun moyen de denoncer des comptes fake !! je passe mon temps a bloquer des personnes qui volent et utilisent les photos d autres comptes et qui drague de manière très agressive !! review faire pour nous proteger de cela !! Pour ne pas changer votre mise à jour BUG!!! cela devient très fatigant car ça se produit à chaque fois depuis presque 1an. |
| | Translated | There is no way to report fake accounts!! I spend my time blocking people who steal and use photos from other accounts and who flirt very aggressively!! how to protect us from that !! To not change your BUG update!!! it becomes very tiring because it has been happening every time for almost 1 year. |
| Review 2 | Original | Depuis quelques temps, les bugs s'enchaînent sur cette appli et ça devient agaçant.. Cette fois-ci, impossible de recharger une page, impossible de voir les messages privées et impossible de se reconnecter sans avoir un message indiquant "une erreur s'est produite". Faites quelque chose ! |
| | Translated | For some time, bugs are linked on this app and it becomes annoying. This time, impossible to reload a page, impossible to see private messages and impossible to reconnect without having a message indicating "an error is produced ". Do something ! |
| Review 3 | Original | Je ne peux même plus liker, reviewer ou m'abonner et lorsque je signale le problème, Instagram ne réponds pas. J'ai bien signaler le problème au moins 4 fois mais rien ne se passe. Le service est très mal organisé. Merci de me remettre les options qui m'ont été prises. |
| | Translated | I can't even like, review or subscribe anymore and when I report the problem, Instagram doesn't respond. I did report the problem at least 4 times but nothing happens. The service is very poorly organized. Thank you for giving me the options that have been taken. |
| Review 4 | Original | Malheureusement, il y a de plus en plus de beugs avec cette application. Que ça soi au niveau des stories ou du partage de photos dans notre fils d'actualité, ça fait déjà la troisième fois (en même pas 1 mois !) que je rencontre des soucis avec Instagram.. |
| | Translated | Unfortunately, there are more and more bugs with this application. Whether it's at the level of stories or sharing photos in our news feed, it's already been the third time (not even 1 month!) That I have encountered problems with Instagram. |

Table A2. Cont.

| | | |
|----------|------------|---|
| Review 5 | Original | Je suis dans l'incapacité de me connecter sur mon compte car le site ne reconnaît pas mon adresse mail. J'ai vérifié son orthographe mais toujours rien. Je me connecte alors en utilisant mon nom d'utilisateur, je suis redirigé vers une page disant qu'une connexion suspecte a été détectée, d'aller sur la page d'aide afin de pouvoir me connecter. Là aucune information ne correspond à mon cas. Je ne peux pas non plus créer un nouveau compte parce que là mon adresse mail est reconnue... |
| | Translated | I am unable to log into my account because the site does not recognize my email address. I checked his spelling but still nothing. I then connect using my username, I am redirected to a page saying that a suspicious connection has been detected, to go to the help page so that I can connect. There no information corresponds to my case. I also cannot create a new account because my email address is recognized there ... |
| Review 6 | Original | j'aime cette application d'Instagram. Je conseil énergiquement de surfer sur ce réseau social...il faut toujours soutenir toutes initiatives qui conduisent à la communication, à tout moment je peux avoir accès à mes ancienne photo, c'est vraiment bien. |
| | Translated | I like this Instagram app. I strongly advise to surf on this social network ... you must always support all initiatives that lead to communication, at any time I can have access to my old photo, it's really good. |
| Review 7 | Original | Ça fait bien 3 ou 4 fois que mon compte est repris ou piraté par un tiers, je ne sais plus me connecter et quand je recherche mon profil il est attribué à quelqu'un d'autre. Impossible de joindre quelqu'un chez Instagram qui pourrait m'aider. M'expliquer pourquoi ça arrive et pourquoi autant de fois? Soit je deviens parano soit il y a une explication... Soit j'arrête Instagram pour de bon. |
| | Translated | It's been 3 or 4 times that my account is taken over or hacked by a third party, I no longer know how to log in and when I search for my profile it is assigned to someone else. Can't reach someone on Instagram who could help me. Explain to me why it happens and why so many times? Either I become paranoid or there is an explanation ... Either I stop Instagram for good. |
| Review 8 | Original | Très déçu de Instagram récemment. Je ne parviens pas à me connecter à mon compte car l'utilisateur n'existe pas. Mais je ne peux pas créer d'autres compte car mon adresse email est déjà utilisée !! veuillez régler ce problème MAJEUR qui m'empêche d'utiliser votre application. |
| | Translated | Very disappointed with Instagram recently. I cannot connect to my account because the user does not exist. But I cannot create other accounts because my email address is already used !! Please resolve this MAJOR problem that prevents me from using your application. |
| Review 9 | Original | Depuis la dernière mise à jour, les conversations privées sont impraticables. Les messages prennent beaucoup plus de place, les images et les gifs prennent absolument toute la page, les déformant, et rendant la conversation illisible et surtout, il est devenu impossible de cliquer dessus pour mieux les voir. Si vous pouviez régler rapidement ce problème car ce n'est pas la première fois qu'il y a des bugs après de MAJ et ça arrive de plus en plus souvent, ça en dit long sur l'intérêt que vous portez à votre appli... |
| | Translated | Since the last update, private conversations have been impractical. The messages take up much more space, the images and gifs take up the whole page, distorting them, and making the conversation unreadable and above all, it has become impossible to click on them to better see them. If you could quickly resolve this problem because it is not the first time that there are bugs after update and it happens more and more often, it says a lot about the interest you have in your app. .. |

Table A2. Cont.

| | | |
|-----------|------------|---|
| Review 10 | Original | Ça fait déjà plusieurs années que je suis sur instagram et je n'ai jamais été insatisfait de ce réseau social. Si je devais vous en recommander un, ce serait celui-ci. Cette application permet de poster de publications d'une manière simple et fluide, de plus nous disposons de filtres assez sympas. Vraiment cool! |
| | Translated | I have been on Instagram for several years now and I have never been dissatisfied with this social network. If I had to recommend one, it would be this. This application allows you to post publications in a simple and fluid way, plus we have pretty cool filters. Really cool! |

References

- Rasool, G.; Ali, A. Recovering Android Bad Smells from Android Applications. *Arab. J. Sci. Eng.* **2020**, pp. 1–27. [CrossRef]
- O'Dea, S. *Global Market Share Smartphone Operating Systems of Unit Shipments 2014–2023*; Technology & Telecommunications: Lanzhou, China, 2020. Available online: <https://www.statista.com/statistics/272307/market-share-forecast-for-smartphone-operating-systems/> (accessed on 12 July 2020).
- My Android Apps. Google, USA, 2020. Available online: <https://play.google.com/apps> (accessed on 20 July 2020).
- The AppInChina App Store Index. AppInChina, China, 2020. Available online: <https://www.appinchina.com/market/app-stores/> (accessed on 27 July 2020).
- Anzhi Market. Anzhi, China, 2020. Available online: <http://www.anzhi.com/> (accessed on 28 July 2020).
- Number of Available Applications in the Google Play Store from December 2009 to December 2019*; Statista: Hamburg, Germany, 2020. Available online: <https://www.statista.com/statistics/266210/number-of-available-applications-in-the-google-play-store/> (accessed on 27 June 2020).
- Pan, Y.; Ge, X.; Fang, C.; Fan, Y. A Systematic Literature Review of Android Malware Detection Using Static Analysis. *IEEE Access* **2020**, *1*. [CrossRef]
- Alazab, M.; Alazab, M.; Shalaginov, A.; Mesleh, A.; Awajan, A. Intelligent mobile malware detection using permission requests and API calls. *Future Gener. Comput. Syst.* **2020**, *107*, 509–521. [CrossRef]
- Taheri, R.; Ghahramani, M.; Javidan, R.; Shojafar, M.; Pooranian, Z.; Conti, M. Similarity-based Android malware detection using Hamming distance of static binary features. *Future Gener. Comput. Syst.* **2020**, *105*, 230–247. [CrossRef]
- Tchakounté, F.; Hayata, F. Supervised Learning Based Detection of Malware on Android. In *Mobile Security and Privacy: Advances, Challenges and Future Research Directions*; Syngress Publishing: Rockland, MA, USA, 2017; pp. 101–154. [CrossRef]
- Tchakounté, F.; Djakene Wandala, A.; Tiguiane, Y. Detection of Android Malware based on Sequence Alignment of Permissions. *Int. J. Comput. (IJC)* **2019**, *35*, 26–36.
- De Lorenzo, A.; Martinelli, F.; Medvet, E.; Mercaldo, F.; Santone, A. Visualizing the outcome of dynamic analysis of Android malware with VizMal. *J. Inf. Secur. Appl.* **2020**, *50*, 102423. [CrossRef]
- Gajrani, J.; Laxmi, V.; Tripathi, M.; Gaur, M.S.; Zemmari, A.; Mosbah, M.; Conti, M. Effectiveness of state-of-the-art dynamic analysis techniques in identifying diverse Android malware and future enhancements. In *Advances in Computers*; Academic Press Inc.: Cambridge, MA, USA, 2020; Volume 119, pp. 73–120. [CrossRef]
- Abdullah, Z.; Muhadi, F.W.; Saudi, M.M.; Hamid, I.R.A.; Foozy, C.F.M. Android Ransomware Detection Based on Dynamic Obtained Features. In *Advances in Intelligent Systems and Computing*; Springer: Berlin/Heidelberg, Germany, 2020; Volume 978 AISC, pp. 121–129. [CrossRef]
- Tang, J.; Li, R.; Wang, K.; Gu, X.; Xu, Z. A novel hybrid method to analyze security vulnerabilities in android applications. *Tsinghua Sci. Technol.* **2020**, *25*, 589–603. [CrossRef]
- Raghuraman, C.; Suresh, S.; Shivshankar, S.; Chapaneri, R. Static and dynamic malware analysis using machine learning. In *Advances in Intelligent Systems and Computing*; Springer: Berlin/Heidelberg, Germany, 2020; Volume 1045, pp. 793–806. [CrossRef]

17. Wang, Y.; Xu, G.; Liu, X.; Mao, W.; Si, C.; Pedrycz, W.; Wang, W. Identifying vulnerabilities of SSL/TLS certificate verification in Android apps with static and dynamic analysis. *J. Syst. Softw.* **2020**, *167*, 110609. [[CrossRef](#)]
18. Alzaylaee, M.K.; Yerima, S.Y.; Sezer, S. DL-Droid: Deep learning based android malware detection using real devices. *Comput. Secur.* **2020**, *89*, 101663. [[CrossRef](#)]
19. Woods, D.W.; Moore, T. Cyber Warranties: Market Fix or Marketing Trick? *Commun. ACM* **2020**, *63*, 104–107. [[CrossRef](#)]
20. Hendriks, F.; Bubendorfer, K.; Chard, R. Reputation systems: A survey and taxonomy. *J. Parallel Distrib. Comput.* **2015**, *75*, 184–197. [[CrossRef](#)]
21. Islam, M.R. Numeric rating of Apps on Google Play Store by sentiment analysis on user reviews. In Proceedings of the 1st International Conference on Electrical Engineering and Information and Communication Technology, ICEEICT 2014, Dhaka, Bangladesh, 10–12 April 2014; Institute of Electrical and Electronics Engineers Inc.: Piscataway, NJ, USA, 2014. [[CrossRef](#)]
22. Mojica Ruiz, I.J.; Nagappan, M.; Adams, B.; Berger, T.; Dienst, S.; Hassan, A.E. Examining the Rating System Used in Mobile-App Stores. *IEEE Softw.* **2016**, *33*, 86–92. [[CrossRef](#)]
23. Genc-Nayebi, N.; Abran, A. A systematic literature review: Opinion mining studies from mobile app store user reviews. *J. Syst. Softw.* **2017**, *125*, 207–219. [[CrossRef](#)]
24. Ehsan, N.; Kelly, L. A survey of utilizing user-reviews posted on Google play store. In Proceedings of the the 29th Annual International Conference on Computer Science and Software Engineering, CASCON '19, Toronto, ON, Canada, 4–6 November 2019; ACM: Toronto, ON, Canada, 2019; pp. 54–63. [[CrossRef](#)]
25. Ureña, R.; Kou, G.; Dong, Y.; Chiclana, F.; Herrera-Viedma, E. A review on trust propagation and opinion dynamics in social networks and group decision making frameworks. *Inf. Sci.* **2019**, *478*, 461–475. [[CrossRef](#)]
26. Alshehri, A.; Marcinek, P.; Alzahrani, A.; Alshahrani, H.; Fu, H. *Puredroid: Permission Usage and Risk Estimation for Android Applications*; ACM International Conference Proceeding Series; Association for Computing Machinery: New York, NY, USA, 2019; pp. 179–184. [[CrossRef](#)]
27. Li, J.; Sun, L.; Yan, Q.; Li, Z.; Srisa-An, W.; Ye, H. Significant Permission Identification for Machine-Learning-Based Android Malware Detection. *IEEE Trans. Ind. Inform.* **2018**, *14*, 3216–3225. [[CrossRef](#)]
28. Xiao, J.; Chen, S.; He, Q.; Feng, Z.; Xue, X. An Android application risk evaluation framework based on minimum permission set identification. *J. Syst. Softw.* **2020**, *163*, 110533. [[CrossRef](#)]
29. Bashir, M.A.; Arshad, S.; Robertson, W.; Wilson, C. Tracing Information Flows Between Ad Exchanges Using Retargeted Ads. In Proceedings of the the 25th USENIX Conference on Security Symposium, SEC'16, Austin, TX, USA, 10–12 August 2016; ACM: Austin, TX, USA, 2016; pp. 481–496.
30. Sun, M.; Wei, T.; Lui, J.C. TaintART: A practical multi-level information-flow tracking system for Android RunTime. In Proceedings of the ACM Conference on Computer and Communications Security, New York, NY, USA, 24–28 October 2016; Association for Computing Machinery: New York, NY, USA, 2016; pp. 331–342. [[CrossRef](#)]
31. Day, M.Y.; Lin, Y.D. Deep learning for sentiment analysis on google play consumer review. In Proceedings of the 2017 IEEE International Conference on Information Reuse and Integration, IRI 2017, San Diego, CA, USA, 4–6 August 2017; Institute of Electrical and Electronics Engineers Inc.: Piscataway, NJ, USA, 2017; pp. 382–388. [[CrossRef](#)]
32. Karim, A.; Azhari, A.; Belhaouri, S.B.; Qureshi, A.A. Machine Learning Algorithm's Measurement and Analytical Visualization of User's Reviews for Google Play Store. *Preprints* **2020**. [[CrossRef](#)]
33. Oyeboode, O.; Alqahtani, F.; Orji, R. Using Machine Learning and Thematic Analysis Methods to Evaluate Mental Health Apps Based on User Reviews. *IEEE Access* **2020**, *8*, 111141–111158. [[CrossRef](#)]
34. Guzman, E.; Maalej, W. How do users like this feature? A fine grained sentiment analysis of App reviews. In Proceedings of the 2014 IEEE 22nd International Requirements Engineering Conference, RE 2014-Proceedings, Karlskrona, Sweden, 25–29 August 2014; Institute of Electrical and Electronics Engineers Inc.: Piscataway, NJ, USA, 2014; pp. 153–162. [[CrossRef](#)]
35. Li, X.; Zhang, B.; Zhang, Z.; Stefanidis, K. A Sentiment-Statistical Approach for Identifying Problematic Mobile App Updates Based on User Reviews. *Information* **2020**, *11*, 152. [[CrossRef](#)]
36. Khalid, H.; Shihab, E.; Nagappan, M.; Hassan, A.E. What do mobile app users complain about? *IEEE Softw.* **2015**, *32*, 70–77. [[CrossRef](#)]

37. Gu, X.; Kim, S. What parts of your apps are loved by users? In Proceedings of the 2015 30th IEEE/ACM International Conference on Automated Software Engineering, ASE 2015, Lincoln, NE, USA, 9–13 November 2015; Institute of Electrical and Electronics Engineers Inc.: Piscataway, NJ, USA, 2015; pp. 760–770. [CrossRef]
38. Nguyen, D.C.; Derr, E.; Backes, M.; Bugiel, S. Short text, large effect: Measuring the impact of user reviews on android app security & privacy. In Proceedings of the IEEE Symposium on Security and Privacy, San Francisco, CA, USA, 19–23 May 2019; Institute of Electrical and Electronics Engineers Inc.: Piscataway, NJ, USA, 2019; pp. 555–569. [CrossRef]
39. Palomba, F.; Salza, P.; Ciurumelea, A.; Panichella, S.; Gall, H.; Ferrucci, F.; De Lucia, A. Recommending and Localizing Change Requests for Mobile Apps Based on User Reviews. In Proceedings of the 2017 IEEE/ACM 39th International Conference on Software Engineering, ICSE 2017, Buenos Aires, Argentina, 20–28 May 2017; Institute of Electrical and Electronics Engineers Inc.: Piscataway, NJ, USA, 2017; pp. 106–117. [CrossRef]
40. Fu, B.; Lin, J.; Liy, L.; Faloutsos, C.; Hong, J.; Sadeh, N. Why people hate your App-Making sense of user feedback in a mobile app store. In Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Chicago, IL, USA, 11–14 August 2013; Association for Computing Machinery: New York, NY, USA, 2013; Volume Part F128815, pp. 1276–1284. [CrossRef]
41. Villarroel, L.; Bavota, G.; Russo, B.; Oliveto, R.; Di Penta, M. Release planning of mobile apps based on user reviews. In Proceedings of the International Conference on Software Engineering, IEEE Computer Society, Austin, TX, USA, 14–22 May 2016; pp. 14–24. [CrossRef]
42. Gao, C.; Zeng, J.; Lyu, M.R.; King, I. *Online App Review Analysis for Identifying Emerging Issues*; Association for Computing Machinery (ACM): New York, NY, USA, 2018; pp. 48–58. [CrossRef]
43. Gao, C.; Wang, B.; He, P.; Zhu, J.; Zhou, Y.; Lyu, M.R. PAID: Prioritizing App Issues for Developers by Tracking User Reviews over Versions. In Proceedings of the 2015 IEEE 26th International Symposium on Software Reliability Engineering, ISSRE 2015, Gaithersbury, MD, USA, 2–5 November 2015; Institute of Electrical and Electronics Engineers Inc.: Piscataway, NJ, USA, 2016; pp. 35–45. [CrossRef]
44. Yu, L.; Chen, J.; Zhou, H.; Luo, X.; Liu, K. Localizing function errors in mobile apps with user reviews. In Proceedings of the 48th Annual IEEE/IFIP International Conference on Dependable Systems and Networks, DSN, Luxembourg, 25–28 June 2018; Institute of Electrical and Electronics Engineers Inc.: Piscataway, NJ, USA, 2018, pp. 418–429. [CrossRef]
45. Hatamian, M.; Serna, J.; Rannenber, K. Revealing the unrevealed: Mining smartphone users privacy perception on app markets. *Comput. Secur.* **2019**, *83*, 332–353. [CrossRef]
46. Tao, C.; Guo, H.; Huang, Z. Identifying security issues for mobile applications based on user review summarization. *Inf. Softw. Technol.* **2020**, *122*, 106290. [CrossRef]
47. Tesfay, W.B.; Booth, T.; Andersson, K. Reputation based security model for android applications. In Proceedings of the 11th IEEE Int. Conference on Trust, Security and Privacy in Computing and Communications, TrustCom-2012-11th IEEE Int. Conference on Ubiquitous Computing and Communications, IUCC-2012, Liverpool, UK, 25–27 June 2012; pp. 896–901. [CrossRef]
48. Chowdhury, N.S.; Raje, R.R. SERS: A security-related and evidence-based ranking scheme for mobile apps. In Proceedings of the 1st IEEE International Conference on Trust, Privacy and Security in Intelligent Systems and Applications, TPS-ISA 2019, Los Angeles, CA, USA, 12–14 December 2019; Institute of Electrical and Electronics Engineers Inc.: Piscataway, NJ, USA, 2019; pp. 130–139. [CrossRef]
49. Kumar, S.; Shukla, S.K. *The State of Android Security*; Springer: Singapore, 2020; pp. 17–22. [CrossRef]
50. Tchakounté, F.; Ndjeumou Ngassi, R. C.; Kamla, V. C.; Udagepola, K. P. LimonDroid: A system coupling three signature-based schemes for profiling Android malware. *Iran J. Comput. Sci.* **2020**, 1–20. [CrossRef]
51. Karim, A.; Azhari, A.; Aldabbas, H.; Alruily, M.; Belhaouri, S.B.; Qureshi, A.A. Classification of Google Play Store Application Reviews Using Machine Learning. *Preprints* **2020**. [CrossRef]
52. Liu, D. Google Play Store Application Scraper. MIT, USA, 2019. Available online: <https://github.com/danieliu/play-scraper> (accessed on 25 July 2020).
53. App Review & Ratings Analysis for Mobile Teams. AppBot, Australia, 2020. Available online: <https://appbot.co/> (accessed on 27 July 2020).
54. Vidas, T.; Votipka, D.; Christin, N. All your droid are belong to us: A survey of current android attacks. In Proceedings of the 5th USENIX Conference on Offensive technologies (WOOT'11), San Francisco, CA, USA, 8 August 2011; p. 10.

55. Tourette, A. Advanced SEO Tool. Alyze, France, 2020. Available online: <https://en.alyze.info/> (accessed on 27 July 2020).
56. Keyword Tool. Keywordtool, Hong Kong, 2020. Available online: <https://keywordtool.io/> (accessed on 27 July 2020).
57. Haryanto, B.; Ruldeviyani, Y.; Rohman, F.; Julius Dimas, T.N.; Magdalena, R.; Muhamad Yasil, F. Facebook analysis of community sentiment on 2019 Indonesian presidential candidates from Facebook opinion data. *Procedia Comput. Sci.* **2019**, *161*, 715–722. [[CrossRef](#)]
58. Miettinen, O.; Steurer, J.; Hofman, A. The Bayes' Theorem Framework for Diagnostic Research. In *Clinical Research Transformed*; Springer Publishing: Berlin/Heidelberg, Germany, 2019; pp. 109–114.
59. Natural Language Toolkit. 2020. Available online: <https://www.nltk.org/> (accessed on 25 July 2020).
60. Raybaut, P. SpiderLib. 2020. Available online: <https://github.com/jromang/spyderlib> (accessed on 27 July 2020).



© 2020 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).