Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 23 March 2020 doi:10.20944/preprints202003.0332.v1

Article

A Security-Related Reputation Scheme of Android Apps based on NLP Analysis of Comments

Franklin Tchakounté[1,*], Athanase Esdras Yera Pagore[2,*], Marcellin Atemkeng[2], Jean Claude Kamgang[3]

[1] Department of Mathematics and Computer Science, Faculty of Science, University of Ngaoundéré; tchafros@gmail.com
[2] Department of Mathematics and Computer Science, Faculty of Science, University of Ngaoundéré; athanase9@gmail.com
[3] Department of Mathematics, Rhodes University, 6140 Grahamstown, South Africa; M.Atemkeng@ru.ac.za
* Correspondence: tchafros@gmail.com (F. T.); athanase9@gmail.com (A. E. Y. P.)

Abstract: Comments are exploited by product vendors to measure satisfaction of consumers. With the advent of Natural Language Processing (NLP), comments on Google Play can be processed to extract knowledge on applications such as their reputation. Proposals in that direction are either informal or interested merely on functionality. Unlike, this work aims to determine reputation of Android applications in terms of confidentiality, integrity, availability and authentication (CIAA). This work proposes a model of assessing app reputation relying on sentiment analysis and text analysis of comments. While assuming that comments are reliable, we collect Google Play applications subject to comments which include security keywords. An in-depth analysis of keywords based on Naive Bayes classification is made to provide polarity of any comment. Based on comment polarity, reputation is evaluated for the whole application. Experiments made on real applications including dozens to billions of comments, reveal that developers lack to make efforts to guarantee CIAA services. A fine-grained analysis shows that not security reputed applications can be reputed in specific CIAA services. Results also show that applications with negative security polarities display in general positive functional polarities. This result suggests that security checking should include careful comment analysis to improve security of applications.

Keywords: Reputation; Android; application; sentiment analysis; comments; security service; NLP; Google Play; polarity

1. Introduction

Due to its open-source nature, Android is the most popular mobile operating system [1]. It is expected to remain popular until 2023 [2]. Android markets, such as Google Play[1] and other third-party markets (AppChina[2], Anzhi[3]), play an important role in the popularity of Android devices. The number of applications has significantly increased from one million in 2013 to about 2.900.000 applications [3] making Google Play a rich place where people can satisfied their needs. Despites Google Play Protect which is exploited for threat protection, there are still malicious applications carefully designed by bad people to impact on user security and privacy [4,5]. Therefore, an urgent need is to develop effective solutions to identify malware. Google provides to consumers, possibility to evaluate submitted applications after deploying for a while. They can give a positive and negative evaluation based on their experience using an application by providing suggestions, opinions and reviews on applications subjects to any complaints, appreciations or suggestions about the functioning of the application. Such opinion comments are useful and relevant to developers and

[1] https://play.google.com/store/apps

[2] https://www.appinchina.co/market/app-stores/

[3] http://www.anzhi.com/

© 2020 by the author(s). Distributed under a Creative Commons CC BY license.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 23 March 2020 doi:10.20944/preprints202003.0332.v1

application store owners to better understand their customers and to recommend improvements [6]. 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 [7]. Tesfay et al. [8] evaluate security-related reputation based on user feedbacks, 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. Some authors rely on review analyses to reveal aspects to improve during application updates [9], [10,11]. Their interest focuses on functionality. Other reputation proposals rely on the risk induced based on permissions requests [12,13], Application Programming Interface (API) calls [9,14] and information flow analysis [15]. However, they require the applications to be installed. Even if they are successful, bad applications would have induced damages. Unlike, the aim behind this research is to prevent bad installations by investigating how to exploit consumer's comments to build the security related reputation of Android applications. By assuming that comments are not fake, this research proposes a reputation model relying on sentiment analysis of opinions and reviews related to security aspects such as confidentiality, integrity, availability and authentication. Experiments on real applications from Google Play reveal that they have bad security related reputations but acceptable functionality related reputations. This document is organized as follows. Section 2 provides a broad overview of authors who conducted similar works. Section 3 presents the research methodology including all aspects for the reputation model. Section 4 presents and discusses results obtained with real applications. The document ends with a conclusion and perspectives.

2. Related Works

According to Genc-Nayebi and Abran [6], 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 analyzed comments 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 [7]. 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 comment 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. Tesfay et al. [8] put in place a private cloud to control user installations of Android applications and to keep track of feedbacks from users. They evaluate reputation of applications based on feedbacks, vendor's reputation and number of users who run the same application. Their proposal is subjective as well as analysis of feedbacks. Our work is more formal with a sentiment analysis coupled to NLP scheme. Reputation can also be evaluated through the risk induced based on permissions requests [12,13,16], API calls [9,14] and information flow analysis [15]. 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. Several prior works analyze reviews to guide vendors and developers on what actions to perform on the applications. Nguyen et al. [9] empirically evaluate incidence of user reviews on updates of Android application security and privacy features such as permissions. They demonstrate that reviews could have been an indicator to code changes to another version. Our proposal can couple the reputation scheme to indicate which security service the developer should concentrate on. Noei et al. [10] investigated significant relationship between reviews and increase of star-ratings. They looked for key topics of user-reviews per category that developers should improve to achieve higher star-ratings. Our work is complement to theirs in that

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 23 March 2020 doi:10.20944/preprints202003.0332.v1

we provide topics specifically on security services, but per application, to be improved by developers. Noei and Lyons [11] further investigate approaches that can help developers and researchers to analyze user-reviews to improve app development process. However, we notice that the studied works consider more functionality aspects in the improvements. In our work, we suggest that security aspects are fundamental, that is why, the reputation based on review analysis is made only on security comments.

3. Reputation Model

The methodology can be summarized in seven essential steps, depicted in the diagram shown in Figure 1. The first step concerns collection of applications based on security-related comments. The second step refers to identification of keywords indicating confidentiality, authentication, integrity and availability flaws. The third step refers to determining polarity of comments based on sentiment analysis. Within the fourth step, comment polarities are classified in security service. The fifth step refers to determine the reputation of an application based on comment polarities from its security-related keywords. Details in each step are provided in the following.

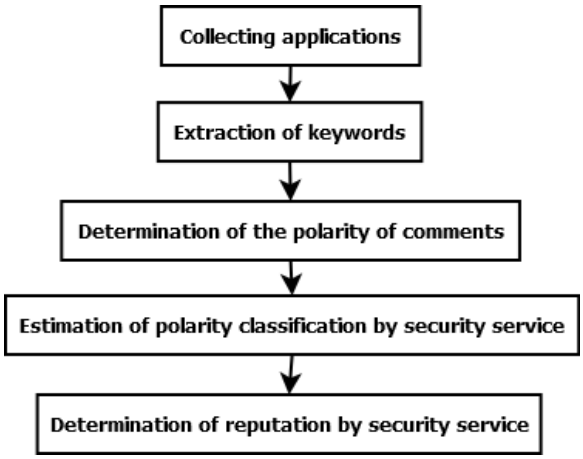

Figure 1. Methodology

3.1. Collecting applications

In this phase, we gather some applications from Google Play based on comments which point out security concerns and other criteria which can influence the reputation of the application. These criteria include user ratings, number of users who downloaded the application and the number of comments made by the customers. We adopt to look for French comments within the scope of this work. Table 1 depicts 10 comments concerning security flaws made on the Instagram application. This example is exploited further on to illustrate the methodology steps.

Table 1. Instagram comments

	Instagram application
Comment 1	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 dragues de manière très agressive !! comment 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.
Comment 2	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 !

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 23 March 2020 doi:10.20944/preprints202003.0332.v1

Comment 3	Je ne peux même plus liker, commenter 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.
Comment 4	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..
Comment 5	Je suis dans l'incapacité de me connecter sur mon compte car le site ne reconnaît pas mon adresse mail. J'ai vérifier 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 à 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...
Comment 6	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.
Comment 7	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.
Comment 8	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 quu m'empêche d'utiliser votre application.
Comment 9	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...
Comment 10	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!

3.2. Extraction of keywords

It involves analyzing the comments identified by highlighting their keywords and in particular those related to security. We adopt to mix French and English keywords within the scope of this work. For this, we use keyword extraction tools such as the Search Engine Optimization (SEO) analysis tool[4], Application TextStat[5] and Appbot[6]. The relevance of these keywords is studied to

[4] https://keywordtool.io/seo-tool

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 23 March 2020 doi:10.20944/preprints202003.0332.v1

bring out those related to the security problem. These keyword extraction tools are based on a global analysis of positive and negative words highlighted in a comment. We proceed to the classification of these keywords by security service as presented in Table 2.

Table 2. Keywords per security service

Security service	Keywords
Confidentiality	Sniffing, Détection, Vol, Intrusion, usurpation, Spoofing, Abus, Malicieux, acharnement, Alert, Confidentiel, Confidentialité, Compromise, fake, volent, protéger, piraté
Authentication	Erroné, Arnaque, Vol, Erreur, Intrusion, usurpation, Passive, Vulnerable, Anomaly, Authentication, Brèche, Certification, Sniffing, Spoofing, volent, protéger, Vulnérable, suspecte
Integrity	Sniffing, Détection, Répudiation, absurdité, Anomaly, fraude, Escroquerie, Mensonge, Arnaqueur, Arnaque, Spam, Malicieux, Acharnement, Vulnérable
Availability	Erroné, Erreur, Bug, Bugger, absurdité, Disponible, Crash, Disponibilité, bloquer, bugs, beugs, impraticables, accès

We justify the selection of these keywords based on their definitions obtained from the Lexicon of computer security and also on the Wikipedia's definitions of security services [17].

Confidentiality: Only authorized persons can have access to the information intended for them (notions of rights or permissions). All unwanted access should be prevented.

Authentication: users must prove their identity by using an access code. Identification and authentication have different terminology; with identification, the user is recognized only by his identifier, while with authentication the user must provide a password. This makes it possible to manage the rights of access to the resources concerned and to maintain confidence in the exchange relationships.

Integrity: the data must be that which is expected and must not be altered accidentally, unlawfully or maliciously. Clearly, the elements considered must be exact and complete.

Availability: access to information system resources must be permanent and flawless during the planned periods of use. Services and resources are accessible quickly and regularly.

3.3. Determination of comment 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 Naïve Bayes. By definition, a Naïve 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 [18], for a characteristic vector $X = (X_1, X_2, \ldots, X_n)$ and a class variable C_k, Bayes' theorem states that:

$$P(C_k|X) = \frac{P(X|C_k)P(C_k)}{P(X)} \qquad k = 1,2,\ldots K \tag{1}$$

where, $P(C_k|X)$ is the posterior probability, $P(X|C_k)$ is the (previous) probability, $P(C_k)$ is the prior class probability (likelihood), $P(X)$ the previous probability of predictor (evidence). Using the chain rule, the probability $P(C_k|X)$ can be broken down as in Equation (2).

[5] https://pypi.org/project/textstat/

[6] https://appbot.co/

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 23 March 2020 doi:10.20944/preprints202003.0332.v1

$$P(C_k|X) = P(X_1, \ldots, X_n \mid C_k) = P(X_1|X_2 \ldots X_n, C_k) \ldots P(X_{n-1} \mid X_n, C_k) P(X_n \mid C_k) \quad (2)$$

We use two environments of analysis for obtaining polarity: the sentiment analysis environment Natural Language Toolkit (NLTK)[7] and the Spider environment[8]. These environments allow sentiment analysis to be 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 comment by relying on the security aspect. Appendix A shows the Python script used to exploit the keywords related to security services. Figure 2 depicts the polarity of each comment from the script. In this specific case, the comment has a negative polarity of 77.9% and a positive polarity of 22.1%. This comment 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 2. Polarity classification of a comment

3.4. Polarity classification by security service

This step determines feelings behind each comment based on polarity and classifying them by security service. Table 3 presents comment polarities for the Instagram application. The comment C2 for example, has a negative feeling with a probability of 0.84.

Table 3. Polarity of comments related to the security of the Instagram application

Comments	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

[7] https://www.nltk.org/

[8] https://github.com/jromang/spyderlib

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 23 March 2020 doi:10.20944/preprints202003.0332.v1

Now we look for classifying polarities by security service. Table 4 shows the breakdown of keywords by security service for comments from a given application.

Table 4. Distribution of security service keywords in comments

	Sentiment	Privacy related words	Integrity related 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

Table 4 shows results about matching comment feelings to security services. Based on Table 4'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.

3.5. Calculation of reputation by security service

At this level, it is a matter of determining the reputation of an application based on the number of comments recorded in Google Play, as well as their respective polarities while taking into account the security service that each comment releases. We obtain the probability that an application has a good or a bad reputation, by subsequently relying on the formula of the probabilistic model [7].

$$E\,(p|\alpha,\beta) = \frac{\alpha}{\alpha+\beta},\ 0 \le p \le 1;\ \alpha,\beta\ > 0 \qquad (3)$$

Where

- α is the total number of positive polarities (N_{pos}) such that $\alpha = \sum \alpha_C + \alpha_I + \alpha_A + \alpha_D$
- β is the total number of negative polarities (N_{neg}) such that $\beta = \sum \beta_C + \beta_I + \beta_A + \beta_D$

With

- 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.

$$\text{Such that } \alpha_i + \beta_i = 1 \qquad (4)$$

Table 5 summarizes different polarity values for each comment in the Instagram application. These values are obtained as follows: The value obtained after implementing the Python script on a given comment, 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

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 23 March 2020 doi:10.20944/preprints202003.0332.v1

above (Table 4). Thus to obtain the values α_i and β_i for a given comment, we recover the number of words of the comment 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., α_i, $\beta_i > 0$). The final polarity value is therefore obtained as a function of the total number of words in the comment highlighting the security aspect and the number of words distributed by security service (Table 4).

Table 5. Determination of (α_i, β_i)

Comments	Polarity index (+/-)	$N_{pos}(\alpha)$				$N_{neg}(\beta)$			
		Confi.	*Integ.*	*Auth.*	*Avail.*	*Confi.*	*Integ.*	*Auth.*	*Avail.*
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_{Av}	β_C	β_I	β_A	β_{Av}

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

- Confidentiality: $E_C = \alpha_c / (\alpha_c + \beta_c) = 0.26$
- Integrity: $E_I = \alpha_I / (\alpha_I + \beta_I) = 0.37$
- Authentification: $E_A = \alpha_A / (\alpha_A + \beta_A) = 0.32$
- Availability: $E_D = \alpha_{Av} / (\alpha_{Av} + \beta_{Av}) = 0.34$

With:

$$\alpha_C = \sum N_{pos}(Conf.i) \; ; \; \alpha_I = \sum N_{pos}(Int.i) \; ; \; \alpha_A = \sum N_{pos}(Aut.i) \; ; \; \alpha_{Av} = \sum N_{pos}(Avail.i)$$

$$\beta_C = \sum N_{neg}(Conf.i) \; ; \; \beta_I = \sum N_{neg}(Int.i) \; ; \; \beta_A = \sum N_{neg}(Aut.i) \; ; \; \beta_{Av} = \sum N_{neg}(Avail.i) \; ;$$

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

$$E(p \mid \alpha, \beta) = Moy(E_C, E_I, E_A, E_D) = 0.32 \quad (5)$$

After the results obtained above, the reputation E that we obtained must be included in the interval [0, 1]. Thus, for an application building a good reputation, the value of E must be greater than the

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 23 March 2020 doi:10.20944/preprints202003.0332.v1

value 0.5; bad for a value less than 0.5 and neutral for a value equal to 0.5. For instance, the Instagram application does not have a good reputation regarding the security aspect. In a more specific context, 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.

4. Results and Discussions

This section presents the main results of our findings. It is subdivided into two points. The first point presents the reputation of Android applications on the security aspect and the second point deals with the functionality aspect.

4.1. Dataset

We gathered 13 applications from Google Play oriented to live messaging, social media, mobile banking and antiviruses. Based on the facts that an application can be subject to hundreds to millions of comments, applications are filtered based on comments with security terms and that their collection is manual, it has been harder to get a lot of applications.

4.2. Reputation based on security

Based on the reputation model, we have obtained the results on the dataset. These results are detailed in Table 6.

Table 6. Reputation in terms of security

Application	**Number of Security related comments**	**Application Polarity**	***E (p | α. β) security***	***E (p | α. β) by security service***			
				Confi.	***Integ.***	***Auth.***	***Avail.***
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
Lecteur de code QR	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	0.54
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

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 23 March 2020 doi:10.20944/preprints202003.0332.v1

BICEC Mobile	5	-	0.41	0.34	0.47	0.45	0.38

Table 6 presents the polarity of the applications on a defined number of comments related to security services. Then, it presents the security related reputation of these applications in general and by security service. Some of them (Lecteur de code QR, 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. Despites 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. "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 7 shows the reputation of the same applications in terms of functionality. 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 comment polarity. 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" application has a bad reputation (15%) meaning that users are satisfied when using this application to perform banking transactions.

Table 7. Reputation in terms of functionality

Applications	**Number of Comments**	**Positive sentiments (%)**	**Negative sentiments (%)**	$E(p \mid \alpha, \beta)$ *functionality*
Instagram	2 288 402	75%	25%	0.75
Facebook	1 479 779	63%	37%	0.63
CIC	1 715	76%	24%	0.76
Messenger	1 010 765	65%	35%	0.65
SGC Connect	10	50%	50%	0.50

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 23 March 2020 doi:10.20944/preprints202003.0332.v1

Whatapp	2 824 936	70%	30%	0.70
Lecteur de code QR	11 026	82%	18%	0.82
Ecobank	643	43%	57%	0.43
Clean Master	139 070	83%	17%	0.83
Security 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

4.4 Reputation based security vs. reputation based functionality

Figure 3 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 Application Programming Interface (APIs) which generates 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 comments reveal that developers really lack to look into security aspects.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 23 March 2020 doi:10.20944/preprints202003.0332.v1

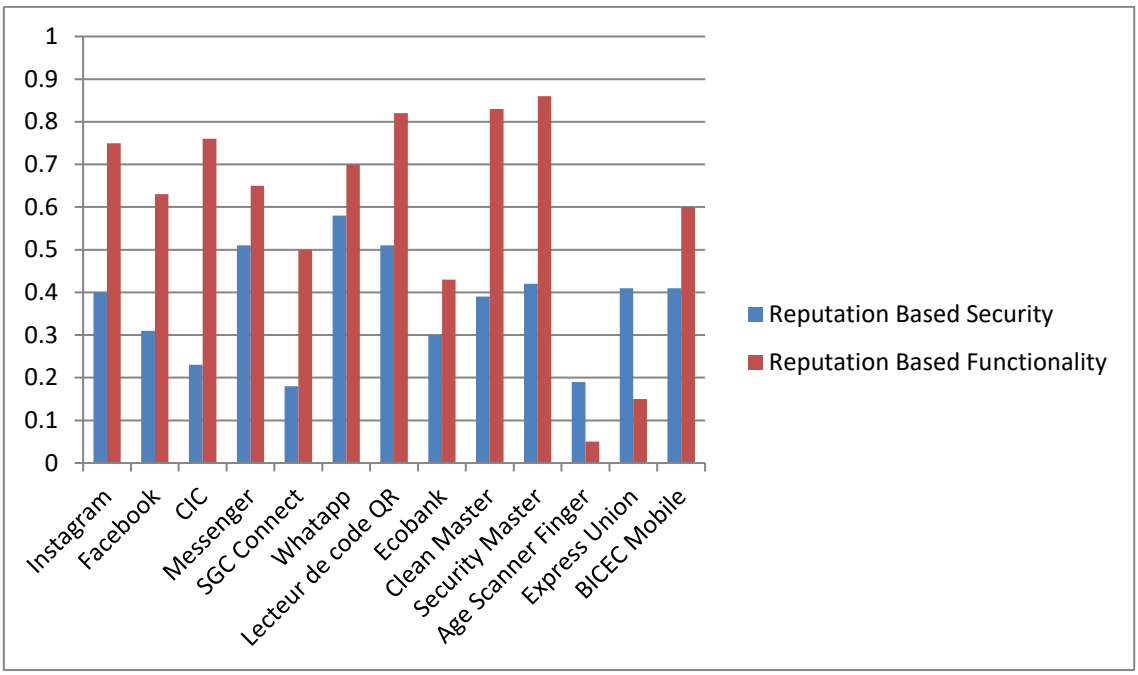

Figure 3. Reputation based security vs. reputation based functionality

4.5. Advantages

The proposed model contributes in some points.

- The model reveals that comments can be effective to indicate security flaws to overcome across updated versions.
- The model is able to prioritize security services in terms of reinforcement in the next version.
- The model is able to provide fine grained security reputations. The model finds through opinions various CIIA 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 certain messaging applications

4.6. Limitations

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

- Retrieving comments on the Google Play is done manually and takes a lot of time;
- This model does not distinguish false comments from real ones. We assumed all comments as true comments;
- The determination of security services for a given comment is not done automatically, but rather manually;
- The consideration of the date of publication of comments 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 comments on the same application through different application stores.

5. Conclusion and Perspectives

We could almost call them "shortcuts", present most of the time on our mobile phones. While some can no longer do without them, others still find it difficult to appropriate them and "stay connected" day and night. Mobile applications are very recent and yet their appearance quickly overwhelmed and completely changed the daily life of the population. Despite its popularity, its

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 23 March 2020 doi:10.20944/preprints202003.0332.v1

momentum, its open-source software and its behavior of programmable structure make it vulnerable to attacks. It was, therefore, a question for us to assess reputation of Android applications based on the opinions and reviews made by users. For this, we have proposed a model based on sentiment analysis and NLP as a solution for determining the reputation of a given application and therefore its potential to be risky or not. Results revealed that the reputation of an application from a functional point of view is not as reliable as that based on the security aspect. This is because an application that meets the security criteria is easier and gives more confidence to its use.

For future works, the model will consider the fact that comments can be done on the same application across different stores and that within the same store, comments can be updated dependently on the developer updates.

Author Contributions: "Conceptualization, Franklin Tchakounté and Athanase Esdras Yera Pagore; methodology, Franklin Tchakounté and Athanase Esdras Yera Pagore; software, Athanase Esdras Yera Pagore; validation, Franklin Tchakounté; writing—original draft preparation, Franklin Tchakounté, Athanase Esdras Yera Pagore and Marcellin Atemkeng; writing—review and editing, Jean Claude Kamgang and Marcellin Atemkeng; supervision, Franklin Tchakounté.; All authors have read and agreed to the published version of the manuscript.".

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

Appendix A

```
# -*- coding: utf-8 -*-
"""
Created on Mon Aug 13 14:30:52 2018
@author: PAGORE
"""
import nltk.classify.util
from nltk.classify import NaiveBayesClassifier
from nltk.corpus import names

def word_feats(words): return dict([(word, True) for word in words])
positive_vocab = [ confiance','débloqué','nécessaire','utilité','util', 'admissible','autorisation','géniale','accessible',
':)' ]
negative_vocab=[   'énervé','piraté','inutil','nullement','inadmissible','déconseiller','ramer','rame','lente','déçu','bu
gs',':(']
neutral_vocab=[  'vraiment','video','le','son','avoir','est','acteur','faire','connaitre','mot','sa','avec','encore','aussi','lu
i','et','on']
positive_features = [(word_feats(pos), 'pos') for pos in positive_vocab]
negative_features = [(word_feats(neg), 'neg') for neg in negative_vocab]
neutral_features = [(word_feats(neu), 'neu') for neu in neutral_vocab]
train_set = negative_features + positive_features + neutral_features
classifier = NaiveBayesClassifier.train(train_set)

# Predict
neg = 0
pos = 0
neu = 0
sentence = "Je rencontre des problèmes aujourd'hui pour visualiser des photos qui ne s'ouvrent pas.... "

sentence = sentence.lower()
words = sentence.split(' ')
for word in words:
    classResult = classifier.classify( word_feats(word))
    if classResult == 'neg':
        neg = neg + 1
    if classResult == 'pos':
        pos = pos + 1
    if classResult == 'neu':
```

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 23 March 2020 doi:10.20944/preprints202003.0332.v1


```
            neu = neu + 1

print('Positive: ' + str(float(pos)/len(words)))
print('Negative: ' + str(float(neg)/len(words)))
print('Neutre: ' + str(float(neu)/len(words)))

print('****************************************')
print('****************************************')

if pos > neg :
    print ('Label : Pos')
if neg > pos :
    print ('Label : Neg')
if neg = pos :
    print ('Label : Neu')
```

References

1. Rasool, G.; Ali, A. Recovering Android Bad Smells from Android Applications. *Arab. J. Sci. Eng.* **2020**, 1–27.

2. S., O. Global market share smartphone operating systems of unit shipments 2014-2023 Available online: https://www.statista.com/statistics/272307/market-share-forecast-for-smartphone-operating-systems/ (accessed on Mar 19, 2020).

3. J., C. Number of available applications in the Google Play Store from December 2009 to December 2019 Available online: https://www.statista.com/statistics/266210/number-of-available-applications-in-the-google-play-store/ (accessed on Mar 19, 2020).

4. Alazab, M.; Alazab, M.; Shalaginov, A.; Mesleh, A.; Awajan, A. Intelligent mobile malware detection using permission requests and API calls. *Futur. Gener. Comput. Syst.* **2020**, *107*, 509–521.

5. Alzaylaee, M.K.; Yerima, S.Y.; Sezer, S. DL-Droid: Deep learning based android malware detection using real devices. *Comput. Secur.* **2020**, *89*, 101663.

6. 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.

7. 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. (Ny).* **2019**, *478*, 461–475.

8. Tesfay, W.B.; Booth, T.; Andersson, K. Reputation based security model for android applications. In Proceedings of the 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; 2012; pp. 896–901.

9. 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

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 23 March 2020 doi:10.20944/preprints202003.0332.v1

Privacy; Institute of Electrical and Electronics Engineers Inc., 2019; Vol. 2019-May, pp. 555–569.

10. Noei, E.; Zhang, F.; Zou, Y. Too Many User-Reviews, What Should App Developers Look at First? *IEEE Trans. Softw. Eng.* **2019**, 1–1.

11. Ehsan Noei; Kelly Lyons A survey of utilizing user-reviews posted on Google play store. In Proceedings of the 29th Annual International Conference on Computer Science and Software Engineering CASCON '19; 2019; pp. 54–63.

12. Alshehri, A.; Marcinek, P.; Alzahrani, A.; Alshahrani, H.; Fu, H. PUREDroid: Permission Usage and Risk Estimation for Android Applications. In Proceedings of the 2019 3rd International Conference on Information System and Data Mining - ICISDM 2019; ACM Press: New York, USA, 2019; pp. 179–184.

13. Arp, D.; Spreitzenbarth, M.; Hübner, M.; Gascon, H.; Rieck, K. Drebin: Effective and Explainable Detection of Android Malware in Your Pocket. In Proceedings of the 2014 Network and Distributed System Security Symposium; 2014.

14. Zimmeck, S.; Wang, Z.; Zou, L.; Iyengar, R.; Liu, B.; Schaub, F.; Wilson, S.; Sadeh, N.; Bellovin, S.M.; Reidenberg, J. Automated analysis of privacy requirements for mobile apps. In Proceedings of the AAAI Fall Symposium - Technical Report; AI Access Foundation, 2016; Vol. FS-16-01-FS-16-05, pp. 286–296.

15. Sun, M.; Wei, T.; Lui, J.C.S. TaintART: A practical multi-level information-flow tracking system for Android RunTime. In Proceedings of the ACM Conference on Computer and Communications Security; Association for Computing Machinery: New York, USA, 2016; Vol. 24-28-October-2016, pp. 331–342.

16. 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. Informatics* **2018**, *14*, 3216–3225.

17. Wikipedia Information security Available online: https://en.wikipedia.org/wiki/Information_security (accessed on Mar 19, 2020).

18. Miettinen, O.S.; Steurer, J.; Hofman, A.; Miettinen, O.S.; Steurer, J.; Hofman, A. The Bayes' Theorem Framework for Diagnostic Research. In *Clinical Research Transformed*; Springer International Publishing, 2019; pp. 109–114.