Is This App Safe for Children? A Comparison Study of Maturity Ratings on Android and iOS Applications

Ying Chen[*], Heng Xu[*], Yilu Zhou[†], Sencun Zhu[*]

[*]The Pennsylvania State University
University Park, PA, USA
{yxc242, hxx4, sxz16}@psu.edu

[†]George Washington University
Washington, DC, USA
yzhou@gwu.edu

ABSTRACT

There is a rising concern among parents who have experienced unreliable content maturity ratings for mobile applications (apps) that result in inappropriate risk exposure for their children and adolescents. In reality, there is no consistent maturity rating policy for mobile applications. The maturity ratings of Android apps are provided purely by developers' self-disclosure and are rarely verified. While Apple's iOS app ratings are considered to be more accurate, they can also be inconsistent with Apple's published policies. To address these issues, this research aims to systematically uncover the extent and severity of unreliable maturity ratings for mobile apps. Specifically, we develop mechanisms to verify the maturity ratings of mobile apps and investigate possible reasons behind the incorrect ratings. We believe that our findings have important implications for platform providers (e.g., Google or Apple) as well as for regulatory bodies and application developers.

Categories and Subject Descriptors

K.4.1: [**Computing Milieux**]: Computers and Society – *Ethics*; H.5.m [**Information Systems**]: Information Interfaces and Presentation (e.g., HCI) – *Miscellaneous*.

General Terms

Measurement, Human Factors

Keywords

Children Safety; Privacy; Maturity Rating; Applications (Apps); Android Apps; iOS Apps; Application Permissions

1. INTRODUCTION

With the rapid adoption of smartphones, tablets, and mobile apps, more and more people use these personal digital devices for communication, entertainment, and professional activities. According to a 2012 survey, approximately half of U.S. mobile consumers own either a smartphone or a tablet [1], and this number will increase to 70 percent by 2013 [2]. The sweeping popularity of smartphones and tablets also affects the user population of children and adolescents. It has been shown that 25% of toddlers used their parents' smartphones in 2011 [3], and 23% of children and teens between the ages of 12 and 17 owned their own smartphones in 2012 [4].

Among smartphone and tablet operating systems, Android and Apple's iOS dominate the U.S. smartphone market by 52.5 and 34.3 percent, respectively [5]. Meanwhile, the growing pace of mobile app offerings is exponential. Approximately 25,000 new apps are added to the Google Play Store per month, amounting to a total of 567,322 apps as of 2012 [6]. There are 18,389 new apps added to the iOS App Store every month, totaling 723,750 apps as of 2012 [7].

In order to help parents determine age-appropriate mobile apps for their children, both Android and iOS apps come with maturity ratings that are similar to the movie and video game industry. Such maturity ratings examine the existence and intensity of mature themes such as mature content, violence, offensive language, sexual content, and drug usage within each app. However, movie and video game industries have official rating organizations such as the Motion Picture Association of America (MPAA) and Entertainment Software Rating Board (ESRB), which set standards for film rating systems – mobile apps do not. Instead of having standard rating rules across platforms, each mobile platform establishes its own rating policy and rating strategy. For example, Android maturity rating policy contains four maturity-rating levels: "Everyone," "Low Maturity," "Medium Maturity," and "High Maturity," while iOS's policy provides four different maturity-rating levels based on the suitable age of audience: "4+," "9+," "12+," and "17+." Both rating systems classify types of objectionable content into four maturity levels, and their classification rules for each level are similar except some minor differences. For instance, apps with intense usage of offensive language are rated as "Low Maturity" (maturity level 2) on Android platform, but they are "12+" (maturity level 3) on iOS.

In terms of *implementing* maturity rating policy, the main difference between iOS and Android platforms is *who* determines or reports the actual ratings. iOS rates each app submitted according to its own policies, but Android's rating system is not as centralized. In fact, a centralized maturity rating system for Android apps' is absent. The maturity ratings for Android apps are purely a result of app developers' self-report. Developers are required to choose one from the four maturity levels before publishing their apps. After submitting to the Google Play Store, an app is available for download in just a few hours. Google does not verify each app's maturity rating unless there are a number of user complaints. The public may raise concerns about the authenticity of the maturity ratings of Android apps, but this requires diligent policing on the part of the end user community. In contrast, iOS has a more strict review process for newly released apps. Apple first requires developers to select from a list of objectionable content and indicate the intensity of the content to generate the maturity rating. According to Apple's "App Store Review Guidelines," Apple examines the contents of apps and adjusts any inappropriate ratings during a review process before the app becomes available to users [8].

WWW 2013, May 13–17, 2013, Rio de Janiero, Brazil.
ACM 978-1-4503-2035-1/13/05.

Due to the laxity of Android's maturity rating policy and the lack of objective judgment of apps' maturity levels provided by developers, many news articles have recognized the drawbacks of Android's rating system. They claim that the Android rating policy is unclear, and it is difficult for developers to understand the difference between the four maturity-rating levels [9]. In addition, according to the Washington Post [10] and recent reports from Federal Trade Commission [11, 12], there is a rising concern among parents who have experienced that the maturity ratings of the apps are unreliable. However, according to our knowledge, little systematic research has conducted to analyze the problems with Android's maturity rating policy and its implementation, not to mention uncovering the risk level of Android apps for children's protection. Therefore, this work is designed to fill this gap.

We contribute the following:

1) We develop a text mining algorithm to automatically predict apps' actual maturity ratings from app descriptions and user reviews.

2) By comparing Android ratings with iOS ratings, we illustrate the percentage of Android apps with incorrect maturity ratings and examine the types of apps which tend to be misclassified.

3) We conduct some preliminary analyses to explore the factors that may lead to untruthful maturity ratings in Android apps.

This paper is structured as follows: In Section 2, we review the relevant work and propose our research questions in Section 3. In Section 4, we describe our methodology for tackling the research questions. The experimental design is presented in Section 5 with our results. We conclude the paper by summarizing our findings and contributions in Section 6.

2. RELATED WORK

Many researchers have studied the risk level of smartphone platforms from the security and privacy perspective [13, 14]. The threats and attacks introduced by Android permission systems have also been analyzed [15, 16]. These studies have found that Android apps normally require more permissions than they actually need in order to support advertisements with the potentially malicious purpose of harvesting users' personal information. Unfortunately, permission warnings make little sense to general public and cannot help users make correct decision before downloading apps. For example, Kelly et al. [17] found that users have poor understandings of what permission disclosures imply and live under the illusion that app marketplaces censor applications in order to reject malicious and low-quality apps. Similarly, Felt et al. [18] found that only 17% of users pay attention to permissions during installation and only 3% of users understand the permission implications comprehensively. Therefore, current permission warnings are both inaccurate and an ineffective means to protect the security and privacy of app users.

Similar to the permissions which are used to measure the security and privacy risks brought by mobile apps, maturity ratings are designed to determine whether there is problematic content in mobile apps for children's protection. In addition, parents highly rely on apps' maturity ratings when choosing apps for their children and adolescents; therefore, an accurate system is essential for assisting them to make correct decisions. Unfortunately, as we already discussed, there is no such system. Apps are assigned incorrect and even conflicting ratings on both Android and iOS platforms [19, 20]. However, to date, no research has systematically uncovered the extent and severity of these unreliable app maturity ratings, nor has any research shown the types of apps that are most often mis-categorized. To bridge this gap, this work develops mechanisms to verify the maturity ratings of mobile apps and investigates the possible reasons behind the incorrect ratings.

3. RESEARCH QUESTIONS

As discussed earlier, there is no standard rating policy for mobile apps. The maturity ratings of Android apps are provided purely by self-report and are rarely verified. While iOS app ratings are considered to be more accurate, they can also be inconsistent with Apple's published policies [9]. Therefore, the first research question is to ascertain:

1. Does iOS rating strictly reflect its policy?

Although iOS's implementation of ratings and its announced policy may be slightly different, Apple's review procedure is still generally accepted as strict and objective. Apple's review guidelines put emphasis on apps' descriptions being relevant to the application content [8]. Therefore, the brief introduction to the content in the apps' description is a good data source for maturity rating prediction. Furthermore, the users who have viewed and run an app may post reviews. Therefore, if Apple's rating is reflected in the description and user review of an app, the maturity ratings can be automatically learned and applied to new apps. Thus, the second research question is raised as:

2. Are app ratings reflected in app descriptions and user reviews? If so, can we build an effective text mining approach to predict the true rating of an app?

We use the maturity ratings provided by iOS as the "truthful maturity ratings." By comparing maturity ratings on iOS and Android, we can further reveal the reliability of maturity ratings on Android. Our third research question is to ascertain:

3. Do Android developers provide accurate maturity ratings for their own apps? For apps published in both markets, are Android ratings consistent with iOS ratings?

If Android developers are found to provide incorrect maturity ratings for their own apps, this study also attempts to identify the factors for the incorrect ratings. Therefore, the last research question is:

4. What are the factors that could lead to untruthful maturity ratings in Android apps in comparison to iOS apps?

4. METHODOLOGY

In order to answer the above research questions, we first analyzed the difference between iOS's rating policy and implementation to assure that iOS's rating scheme can be used as a baseline for verifying the reliability of Android's ratings. Based on the iOS's ratings, a text-mining algorithm ALM was developed by analyzing the app descriptions and users' reviews in order to predict the maturity ratings of apps.

4.1 iOS Maturity Rating Policy vs. Implementation

As shown in Table 1, the maturity rating policy of iOS [21] contains four levels based on user's age: "4+", "9+", "12+", and

"17+". Its rating policy describes four categories of mature content: violence, offensive language, sex, and other. To clearly identify the categories, we manually categorized iOS maturity rating policy and each category was given an abbreviation:

- The *violence* category includes cartoon/fantasy violence (*A*) and realistic violence (*B*).
- The *sex* category includes suggestive themes (*D*) and sexual content (*E*).
- The *offensive language* category includes profanity and crude humor (*F*).
- The *other* category includes drug/alcohol/tobacco usage (*G*) and simulated gambling (*H*).

In Table 1, we also assign "*1*" or "*2*" to indicate the intensity or frequency of the above mentioned harmful category: "*1*" denotes mild/infrequent appearance, and "*2*" denotes the intense/frequent appearance. For example, *A1* means mild/infrequent appearance of cartoon and fantasy violence, while *E2* means intense/frequent appearance of sexual content.

During the implementation of its policy, iOS provides detailed reasons for each maturity rating. For example:

Rated 9+ for the following:

- *Frequent/Intense Cartoon or Fantasy Violence*

Apps may be rated to a specific maturity level for containing a single type or multiple types of objectionable content. For apps rated for containing a single type of objectionable content, the type of objectionable content becomes a "dominant reason" for the maturity level. For example, there are apps rated as "17+" only for containing "frequent/intense sexual content or nudity (*E2*)", *E2* is determined as a dominant reason for maturity level "17+".

The dominant reasons for each rating level are selected by a bottom-up search from maturity rating 9+ to rating 17+ in actual app ratings.

Our data source was the total of 1,464 iOS apps described in Section 5.1. First, we identify all the dominant reasons cited for rating 9+, including *A1, A2, B1, C1, D1,* and *F1*. These objectionable contents become unessential reasons for rating 12+ and rating 17+. By removing the unessential reasons from 12+ apps, some of apps, previously containing multiple types of objectionable content, now only contain one type of objectionable content. Similarly, the dominant reasons for rating 12+ can be determined, including *B2, C2, E1, F2, G1, H1, H2*. Lastly, the dominant reasons for 17+ can be determined as *D2, E2, G2*. Table 2 summarizes the findings from actual apps with cited reasons for each rating. We find that an app's maturity rating does not boost to the next level even if it contains all dominant reasons of the lower levels. Thus, the dominant reasons are necessary and sufficient. By comparing Table 1 and Table 2, we can conclude that Apple's actual rating policy is quite different from its official rating policy.

By comparing iOS official rating policy and actual rating practice, we find the following main differences:

Violence category: 1) The reason "frequent/intense cartoon, fantasy violence" (*A2*), listed in both 12+ and 17+ in the iOS official policy, leads to 9+ in actual ratings. 2) The reason "frequent/intense horror themes" (*C2*), listed in rating 17+ in the iOS official policy, leads to 12+ in actual ratings.

Table 1 Apple iOS official maturity rating policy

Maturity levels	Violence	Sex	Offensive language	Other
4+	-	-	-	-
9+	Mild/infrequent cartoon, fantasy (*A1*) or realistic violence (*B1*), or infrequent/mild horror themes (*C1*)	Infrequent/mild mature, suggestive themes (*D1*)	-	-
12+	Frequent/intense cartoon, fantasy (*A2*) or realistic violence (*B2*)	Mild/infrequent mature or suggestive themes (*D1*)	Infrequent mild language (*F1*)	Simulated gambling (*H1,H2*)
17+	Frequent/intense cartoon, fantasy (*A2*) or realistic violence (*B2*), Frequent/intense horror themes (*C2*)	Frequent/intense mature and suggestive themes (*D2*), Sexual content, nudity (*E1,E2*)	Frequent/ intense offensive language (*F2*)	Alcohol, tobacco, drugs (*G1,G2*)

Table 2 Apple iOS actual maturity rating policy derived from reasons Apple cited to rate each app

Maturity levels	Violence	Sex	Offensive language	Other
4+	-	-	-	-
9+	Cartoon, fantasy violence (*A1, A2*), Infrequent/mile realistic violence (*B1*), Infrequent/mile horror/fear themes (*C1*)	Infrequent/mile mature and suggestive themes (*D1*)	Infrequent/mild profanity or crude humor (*F1*)	-
12+	Frequent/intense realistic violence (*B2*), Frequent/intense horror/fear themes (*C2*)	Infrequent/mile sexual content, nudity (*E1*)	Frequent/intense profanity or crude humor (*F2*)	Infrequent/mile alcohol, tobacco, drugs use or references (*G1*), simulated gambling (*H1, H2*)
17+	-	Frequent/intense mature and suggestive themes (*D2*), Frequent/intense sexual content, nudity (*E2*)	-	Frequent/intense alcohol, tobacco, drugs use or references (*G2*)

Table 3 Maturity rating policy for Android applications

Maturity levels	Violence	Sex	Offensive language	Other	Social feature	Location
Everyone	-	-	-	-	-	-
Low maturity	Mild cartoon, fantasy violence (*A1*)	-	Potentially offensive content (*F1*)	-	Some social features but not allow user to communicate (*I1*)	Collect for service (*J1*)
Medium maturity	Intense fantasy (*A2*) or realistic violence (*B2*)	Sex reference (*E1*)	Profanity or crude humor (*F2*)	References to drug, alcohol and tobacco use (*G1*), and simulated gambling (*H1*)	Social features allow user to communicate (*I2*)	Collect for sharing (*J2*)
High maturity	Graphic violence (*B3*)	Frequent instances of sexual (*D2*), and Suggestive content (*E2*)		Strong alcohol, tobacco, drug (*G2*), and Strong simulated gambling (*H2*)	Social features allow user to communicate (*I2*)	Collect for sharing (*J2*)

Offensive language category: 1) The reason "infrequent/mild language" (*F1*), listed in 12+ in the iOS official policy, leads to 9+ in actual ratings. 2) The reason "frequent/intense offensive language" (*F2*), listed in rating 17+ in the iOS official policy, leads to 12+ in actual ratings.

Sex category: The reason "infrequent/mild sexual and nudity content" (*E1*), listed in rating 17+ in the iOS official policy, leads to 12+ in actual ratings.

Other category: The reason "infrequent/mild alcohol, tobacco, drug use" (*G1*), listed in rating 17+ in the iOS official policy, leads to 12+ in actual rating.

Based on this analysis, we can see that iOS actually downgrades its official maturity policy during implementation. The inconsistency between iOS official policy and its actual ratings could cause problems. When parents view an app's description page at iOS store, they may be misled. Parents who intend to choose apps with maturity rating 12+ for their children to avoid exposure to horror content, frequent offensive language, and sexual/nudity content may actually get an app that contain all aspects of such undesirable content. The only avenue to avoid this situation is to read through all the reasons, which requires significant effort on the part of the parent. Yet, parents are frequently unaware of the discrepancies between the actual maturity rating and the official policy and instead trust the actual maturity ratings as they are listed.

4.2 Android Apps' Maturity Ratings

As presented in Table 3, we manually categorized Android maturity rating policy. Android has its own rating policy with four levels [22]: "Everyone", "Low Maturity", "Medium Maturity", and "High Maturity". Compared to iOS maturity rating policy, Android's policy contains two additional categories (i.e., social feature and location) with five additional tokens. The additional tokens are: the social features that disallow users to communicate (*I1*) and the social features that allow users to communicate (*I2*); collecting user locations for service (*J1*), and collecting user locations for sharing (*J2*); the token value "*3*" represents the graphic appearance of violence content.

Table 3 presents the basic rules for differentiating Android apps' maturity levels. The rating of "Everyone" means there is no harmful content. The rating of "Low Maturity" means that violent content, offensive language, social feature, and collection of location information may appear, but are mild and infrequent with minimal effect on children's mental health and privacy. With the rating of "Medium Maturity", all six categories of objectionable content (i.e., violence, offensive language, sex, other, social feature and location.) may appear intensely and frequently with the exception of sex content, alcohol/tobacco/drug, and gambling content which are illegal for minors under 18 to view. Apps belonging to the level of "Medium Maturity" are arguably harmful for children under 13 years old for viewing or engaging. Finally, the highest maturity level – "High Maturity" contains content for adults only such as significant sexual and violent content (see Table 3). It seems that Android's maturity rating policy is reasonable and clear. However, the reliability of Android's actual ratings by developers remains a question because the actual ratings largely rely on developers' comprehension and assessment of the policy.

Our comparison of Android's maturity rating policy (Table 3) with the iOS's actual maturity rating policy (Table 2) reveals that both platforms categorize maturity ratings into four levels: "Everyone", "Low Maturity", "Medium Maturity", and "High Maturity" in Android; and "4+", "9+", "12+", and "17+" in iOS.

Table 4 Comparison of Andriod's Policy and iOS's Policy

Maturity levels	Maturity rating levels by Android	Maturity rating levels by iOS
1	Everyone	4+
2	Low maturity	9+
3	Medium maturity	12+
4	High maturity	17+

Because the maturity levels for content in each category are mostly similar (see Table 4), we argue that iOS's maturity rating scheme is reasonably reflected in that of Android's maturity rating scheme, except the following discrepancies:

- Android does not consider horror content (*C*) as mature content, while iOS does include horror content (*C*) as mature content.
- Android considers graphic violence (*B3*) as mature content while iOS directly rejects apps with graphic violence.
- Android integrates privacy protection in its maturity rating policy by including the social feature (*I*) and location collection (*J*). However, no corresponding privacy-related consideration exists in the maturity rating scheme by iOS.

- Frequent/intense cartoon violence and fantasy violence (*A2*) is rated as "Medium Maturity" (i.e., level 3) in Android but as "9+" (i.e., level 2) in iOS.
- Frequent/intense simulated gambling (*H2*) is rated as "High Maturity" (i.e., level 4) in Android but is rates as "12+" (i.e., level 3) in iOS.

Thus, to use iOS actual maturity rating as a baseline for measuring the reliability of self-reported maturity ratings on the Android platform, we had to exclude those cases in which maturity ratings contain the above schemes reflecting the discrepancies between iOS and Android's rating policies. After such exclusions, Android's policy and iOS's actual rating scheme should be the similar for all the remaining maturity ratings. Thus, we can now use iOS actual maturity rating as a baseline to examine the reliability of Android apps' maturity ratings.

4.3 Comparing Apps on iOS and Android

After establishing the baseline for evaluation, we match apps from Google Play with those same apps on the iOS App Store. The index scheme for each app on iOS and Android is different (each Android app has a unique package name that serves as the application ID; and each iOS app has a unique Apple ID). However, apps' names are often consistent across the platforms of iOS and Android for branding purpose.

We used a program to automatically search the iOS App Store based on apps' names collected from Google Play. It reveals that the same app for both platforms could have slightly different names. Thus, for each Android app, we choose up to 150 search results from the iOS App Store. For those showing similar app names, we conducted analysis to determine the closest fit.

Algorithm: CalculateEditDistance
Input: Android app name $N_{Android}$, results returned by iTune N_k
Output: The edit distance, Ed_k, of, $N_{Android}$, and, N_k.
$Ed_{k1} = SearchDifferentWords\ (N_{Android}, N_k)$
$Ed_{k2} = SearchDifferentWords\ (N_k, N_{Android})$
RETURN $min(Ed_{k1}, Ed_{k2})$

FUNCTION *SearchDifferentWords* $(N_{Android}, N_k)$
  Split, $N_{Android}$, into a word set, $W_{Android} = \{a_1, a_2, \cdots, a_i\}$
  Split, N_k, into a word set, $W_k = \{b_1, b_2, \cdots, b_j\}$
  FOR each, $a_i \in W_{Android}$
    IF $\exists b_j \in W_k$, equals or only one letter different from, a_i
      Delete, b_j, from, W_k
    END IF
  END FOR
RETURN number of words left in, W_k
END

Specifically, to find the iOS app whose name is mostly similar to an Android app, we use minimum edit distance between the Android app name and each returned iOS app name to estimate the similarity of two names (see the above algorithm). For an Android app, denote its name by $N_{Android}$. For each $N_{Android}$, the search result from the App Store produces a set, $S = \{N_1, N_2, \cdots, N_k, \cdots, N_{result\_size}\}$ $(0 < result\_size < 150$. For each N_k, the algorithm *CalculateEditDistance* returns the name N_{iOS} which is the minimum edit distance to $N_{Android}$ in S.

To confirm that a pair of iOS app and Android app with similar names is the same app, their descriptions and developers' company names were further compared. Finally, the confirmed similar apps' icons and screenshots were visually compared by two individual researchers to ensure that these two apps in Android and iOS were the same. The selected app pairs were then used as our comparison dataset.

4.4 ALM—Automatic Label of Maturity Ratings for Mobile Apps

If an Android app has an iOS version, the algorithm in the previous subsection is sufficient to identify the same app on iOS. However, not all Android apps have a counterpart in the iOS App Store. We label these apps as "Android-only" apps. For Android-only apps, we need to determine their actual maturity rating (not based on the ratings provided by their developers). Thus, we propose a text-mining-based Automatic Label of Maturity ratings (ALM) algorithm. ALM is a semi-supervised learning algorithm, and it processes apps' descriptions and user reviews to determine maturity ratings. The more technical detail of ALM is described below.

4.4.1 Building seed-lexicons for objectionable content detection

For the iOS apps in a training dataset, we group their descriptions according to the contained objectionable contents. Apps containing only one type of the objectionable content are organized based on their rating scheme together with their corresponding token, such as *A1.txt*, *A2.txt*, *B1.txt*, and *H2.txt*. For example, the *A1.txt* file contains the descriptions and users' reviews of all the apps whose maturity ratings are "9+" caused by "infrequent/mild cartoon and fantasy violence" (A1).

Human experts read grouped app descriptions and select seed lexicons to detect objectionable content. This manual labeling procedure produces the seed-lexicons for each mature content category. Human experts are then asked to extract as many terms as possible. Another approach is to use unsupervised learning algorithms to extract seed-terms automatically. However, the performance of this approach may suffer. Thus we argue that it is reasonable to manually select seed-lexicons to generate accurate classification result in this study.

After the seed-terms are generated for each type of objectionable content, they are grouped into three bigger lexicons denoted as T_i, $i \in 9, 12, 17$ for classifying the maturity rating: 9+, 12+, and 17+ (as shown in Table 5). T_i presents the objectionable contents for each level i, and it only includes nouns, verbs, adjectives, and adverbs in this study.

Table 5 Group seed-lexicons for classification

Grouped Lexicon	Seed-lexicons
17+	*D2*, *E2*, *G2*
12+	*B2*, *C2*, *E1*, *F2*, *G1*, *H1*, *H2*
9+	*A1*, *A2*, *B1*, *C1*, *D1*, *F1*

4.4.2 Assigning initial weights to seed-terms

Next we assign initial weights for the terms in the seed-lexicons. To do so, positive instances and negative instances are separately grouped into sets for each maturity level, as shown in Table 6.

Table 6 Positive and negative instances for maturity classification

Maturity levels	Positive instances set	Negative instances set
17+	Apps rated 17+ by iOS	Apps rated 4+, 9+, 12+ by iOS
12+	Apps rated 12+ by iOS	Apps rated 4+, 9+ by iOS
9+	Apps rated 9+ by iOS	Apps rated 4+ by iOS
4+	Apps rated 4+ by iOS	-

The set of positive instances and negative instances are denoted as P_i and N_i respectively for level i $\in 9, 12, 17$. For each seed-term t $\in$ T_i, denote its frequency in P_i and N_i as t_p and t_n, respectively. Therefore, the initial weight of t can be calculated using Equation (1).

$$w_t = \begin{cases} t_p & if\ t \in P_i \backslash N_i, \\ -t_n & if\ t \in N_i \backslash P_i, \\ \frac{t_p}{t_n} & if\ t \in P_i \cap N_i, \\ 0 & if\ t \notin P_i \cup N_i. \end{cases} \quad (1)$$

4.4.3 Classification

Once the seed-terms and their weights are generated, we can calculate apps' maturity ratings. For each app a, all terms in its description are selected and categorized as a set $A = t_k$. We further define a random threshold α to differentiate positive instances with negative instances. The value of α does not affect the result, because in the training phase, the *Expand_Adjust* algorithm (described in the next section) will adjust term weights to fit the threshold, and later the adjusted weights and the threshold are used together to calculate the maturity rating of test instances. Thus, for app a, its maturity rating m_a can be determined by Equation (2) and Equation (3).

$$s_i^a = \sum_{t_j \in T_i} (t_j * w_{t_j}) \quad (2)$$

$$m_a = \begin{cases} 17+ & if\ s_{17}^a > \alpha \\ 12+ & if\ s_{17}^a < \alpha,\ s_{12}^a > \alpha \\ 9+ & if\ s_{17}^a, s_{12}^a < \alpha,\ s_9^a > \alpha \\ 4+ & otherwise \end{cases} \quad (3)$$

4.4.4 Expanding seed-lexicons and adjusting weights

Through the human-assisted process above, we found that the classification accuracy in determining apps' maturity ratings with only the seed-lexicons and their initial weights is around 70%. This is because seed lexicons can only partially reflect objectionable content. However, other terms that appeared frequently in the positive instances should also be added to the lexicon set to further improve the accuracy of classification. In addition, weights of terms should be further adjusted to suit the content. Therefore, we further use the unsupervised learning algorithm *Expand_Adjust* to add frequent terms in the positive instances into consideration, and our algorithm automatically adjusts the weights for both seed terms and non-seed terms, to find an optimal balance of precision and recall in the classification.

All terms in $P_i \cup N_i \backslash T_i$, $i \in 9, 12, 17$, are categorized into the non-seed-terms set T_{ns}, and initial weights of the terms in T_{ns} are 0. As the auto-labeling algorithm runs, instances may be mis-classified. Therefore, sets F_p and F_n are denoted to present the false positive and false negative set, respectively.

Once the terms and weights are optimized, apps' maturity ratings can be estimated by the classification algorithm described in the previous subsection.

Algorithm: Expand_Adjust
Input: Positive instance set P, negative instance set N, false positive set F_p, false negative set F_n, weights of all terms: $W = \{w_t\}$, seed-term set T_s, non-seed-term set T_{ns}.
Output: Updated weights of all terms: $W = \{w_t\}$
WHILE (the size of F_p and F_n can further be decreased)
  W = DecreaseFN (T_s , W);
  W = DecreaseFP (T_s , W);
  W = DecreaseFN (T_{ns} , W);
  W = DecreaseFP (T_{ns} , W);
END WHILE

FUNCTION DecreaseFN ($Term\ set\ T_0$, W)
  WHILE (size of F_n can further be decreased)
    FOR ($t \in T_0$)
      Find max (w_t) which can decrease size of F_n, but not increase size of F_p)
    END FOR
  END WHILE
  RETURN W
END
FUNCTION DecreaseFP ($Term\ set\ T_0$, W)
  WHILE (size of F_p can further be decreased)
    FOR (each $t \in T_0$)
      Find min (w_t) which can decrease size of F_p, but not increase size of F_n)
    END FOR
  END WHILE
  RETURN W
END

5. EXPERIMENT

In this section, we describe our experimental dataset, design and results.

5.1 Data Collection

An automatic crawler was built to collect data from the Google Play Store. The crawler ran for a week from 9/26/12 to 10/2/12, and collected two datasets. The first dataset was a pretest dataset which contained metadata from 1,000 Android apps -- top 500 paid apps and top 500 free Android apps from all categories on the Google Play Store. This dataset was used to conduct an initial assessment on the types of apps that frequently received incorrect maturity ratings. Using the iOS app counterparts, our result showed that the category of "Games" received the most incorrect maturity ratings (see Fig.1). Given that the category of "Games" was shown to be the most popular category for app download [23], our second round of data collection focused only on Android apps in the category of "Games".

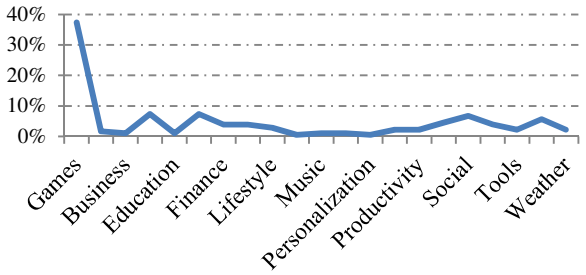

Figure 1 Distribution of apps with incorrect maturity ratings by different categories from dataset 1

The second dataset (main dataset) contained the metadata and user reviews crawled from 5,059 apps in the category of "Games". Metadata included a rich spectrum of information such as app package name (a unique id for each Android app), app name, developer's name, developer's company, developer's website, category, price, currency, number of installations, icon, screenshot, permission, and description. The collected apps were equally distributed among 8 different subcategories of games: arcade & action, brain & puzzle, cards and casino, casual, live wallpaper, racing, sports games, and widgets. A total of 729,128 user reviews were collected, resulting in 144 reviews per app on average.

For each Android app, we searched through iOS App Store using the method descripted in section 4.3. A total of 1,464 apps were found on iOS App Store and the rest 3,595 apps were classified as Android-only apps.

5.2 Experiment 1: Predicting Apps' Maturity Ratings by the ALM algorithm

For Android-only apps, we used the ALM algorithm described in Section 4.4 to automatically label maturity ratings. In this experiment, the 1,464 apps which are available on both Android and iOS were used as the training set, and the 3,595 Android-only apps were used as the testing set.

We conducted a 10-fold classification on the training set. Standard evaluation metrics for classification, precision, recall, and f-score were used as our evaluation metrics. In particular, precision presents the percent of identified positive instances that are truly positive instances. Recall measures the overall classification correctness, which represents the percent of actual positive instances that are correctly identified. F-score represents the weighted harmonic mean of precision and recall, which is defined as:

$$f-score = \frac{2(precision * recall)}{precision + recall} \qquad (4)$$

The performance of the ALM algorithm on the training set is presented in Table 7. ALM achieved high precision in maturity rating detection across all maturity levels. It performed extremely well in detecting maturity ratings of "17+" and "4+". This is intuitive because apps with high maturity rating normally contain extreme mature content, while apps with low maturity rating often do not contain any mature content. Therefore, it seems reasonable to conclude that ALM is most effective and less error prone in detecting extreme cases. For apps with maturity rating "12+" and "9+", ALM's performance was slightly lower due to the infrequent and subtle mature content.

Table 7 Detection result of ALM on training set

Maturity levels	# of positive instances	# of negative instances	# of seed terms	# of expanded terms	Precision	Recall	F-score
17+	31	1,433	48	67	100%	100%	100%
12+	229	1,204	176	282	96.6%	99.6%	98.1%
9+	155	1,049	134	235	93.9%	99.4%	96.6%
4+	1,049	0	0	0	99.8%	98.4%	99.1%

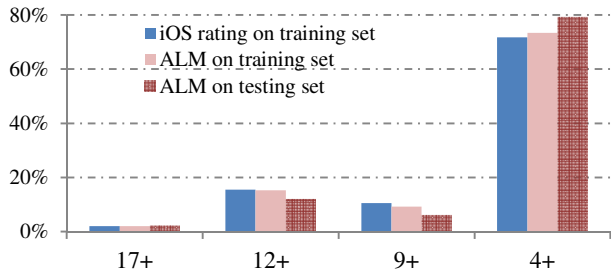

Figure 2 Distributions of apps in training and testing set

Once we verified that ALM performed effectively on training set, we applied the same model to the testing set and compared the distributions of the detection results between the training and testing sets (as Fig. 2). We observe that the distributions of the two sets are similar. Thus, it is reasonable to use ALM to predict true maturity ratings of an app. With ALM, the maturity ratings for Android-only apps can also be verified. Currently, Apple takes a long time to manually rate every single app submitted by third party developers. If this labor-intensive procedure can be assisted by automatic rating algorithms such as ALM, the app maturity rating process for iOS apps can be significantly shortened.

5.3 Experiment 2: Overrated and Underrated Android Applications

For any Android app, a, its maturity rating provided by the developer, is denoted as m_a^A. Define the maturity level l_a^A as:

$$l_a^A = \begin{cases} 1 & if\ m_a^A = "Everyone", \\ 2 & if\ m_a^A = "Low\ Maturity", \\ 3 & if\ m_a^A = "Medium\ Maturity", \\ 4 & if\ m_a^A = "High\ Maturity". \end{cases} \qquad (5)$$

Similarly, the actual maturity rating for the app a is denoted as m_a^i, where m_a^i is the maturity rating on iOS if the app is available on Apple Store, or it is the predicted maturity rating by ALM otherwise. Then its maturity level l_a^i can be expressed as:

$$l_a^i = \begin{cases} 1 & if\ m_a^i = "4+" \\ 2 & if\ m_a^i = "9+" \\ 3 & if\ m_a^i = "12+" \\ 4 & if\ m_a^i = "17+" \end{cases} \qquad (6)$$

Therefore, if $l_a^A > l_a^i$, the app a is overrated and the overrating level is $l_a^A - l_a^i$. If $l_a^A < l_a^i$, the app a is underrated and the underrating level is $l_a^i - l_a^A$. In our dataset 2, among the 1,464 apps that were available on both Android and iOS, 265 apps

(18.1%) were overrated (i.e., their maturity ratings on Android were higher than on iOS), and 142 apps (9.7%) were underrated (i.e., their maturity ratings on Android were lower than on iOS).

5.3.1 Overrated Android Applications

Of the 265 overrated Android apps, there were 4 apps (1.5%) with an overrating level of 3, which means those apps were rated as "High Maturity" on Android but only rated as "4+" on iOS. In addition, there were 46 Android apps (17.4%) with an overrating level of 2, and 215 Android apps (81.1%) with an overrating level of 1.

Next, we discuss some possible reasons that could be counted for the overrating phenomenon.

Intelligence. Android's self-reporting system requires developers to fully comprehend its rating policy. Many Android apps are overrated because developers are under the illusion that the maturity rating is also the criterion to judge users' capabilities or intelligence levels. For example, a chess game is rated as "Medium Maturity" but not "Everyone", because the developer may think that children younger than 12 year-old are not capable of playing the chess game. Because of this reason, many games in the subcategory "Brain & Puzzle" are overrated. Similarly, developers may also think that maturity ratings should reflect users' capability to complete some tasks such as wearing makeup, making cakes, taking care of pets, decorating houses, constructing cities, and running businesses. Android apps with these types of contents are often overrated.

Simulated Gambling. As discussed earlier, inconsistencies exist between the maturity rating policies of Android and iOS. One of these inconsistencies lies in which maturity level simulated gambling should belong to. According to iOS maturity ratings, casino games do not necessarily involve gambling, such as card games, bingo, bridge, backgammon, coin games, mahjong, slots, domino, poker, and etc. However, once an app requires players to bid, or to play for real money, it becomes a simulated gambling. Many Android apps are overrated for this reason.

Violence. As in the case of gambling, developers are also easily confused in determining the existence of violence content in their games. Normally they get tripped up on determining whether or not the following content is considered as violent content: gun shooting, cannon shooting, hunting, racing, and attacking territories. In situations under which developers are not sure about maturity levels for these contents, they tend to overrate the apps.

Mature and Suggestive Themes. The last item causing confusion for developers is the definition of mature and suggestive themes. For example, is "dating" suggestive? Is the term "boyfriend/girlfriend" suggestive? How about "nightclub"? In these circumstances, developers are easily confused in determining maturity levels and thus tend to overrate the apps.

The distribution of overrated Android apps is shown in Fig. 3. To alleviate the overrating problem, we suggest that Android maturity rating policies clearly define the meaning of maturity ratings as an indicator of harmful content for children or adolescents. Maturity rating should not reflect users' capabilities or intelligence levels as it causes undue confusion about ratings. In addition, Android maturity policy should provide clearer definitions and detailed explanations about the meanings of "simulated gambling", "violence", and "mature and suggestive themes", to guide developers to correctly rate their Android apps.

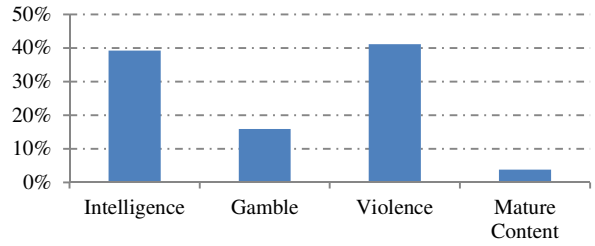

Figure 3 Distribution of overrated apps

5.3.2 Underrated Android Applications

Among those 142 Android apps that were underrated, 36 apps (25.4%) were underrated by 2 levels, and 106 apps (74.6%) were underrated by 1 level. Among the apps underrated by 2 levels, only 1 app was underrated from "17+" to "Low Maturity"; and 35 apps were underrated from "12+" to "Everyone". Among the apps underrated by 1 level, 5 apps were underrated from "17+" to "Medium Maturity"; 25 apps were underrated from "12+" to "Low Maturity"; and 76 apps were underrated from "9+" to "Everyone".

As shown in Fig. 4, most apps underrated by 2 levels contained content such as alcohol, tobacco, drug, and gamble; while apps underrated by 1 level often contained cartoon, fantasy violence, mature content, and suggestive themes.

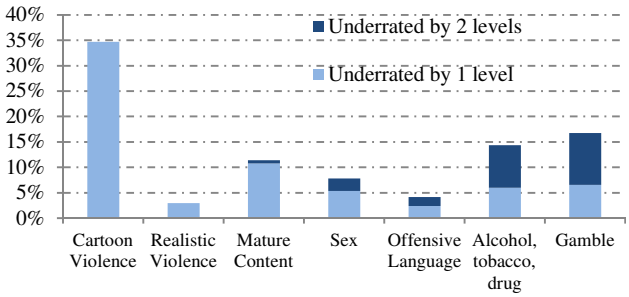

Figure 4 Distribution of underrated apps

Unlike overrated apps, the underrated apps may directly harm children's mental health, because those apps conceal their actual maturity levels to parents and minors. As shown in Figure 4, the underrated apps might contain violent content, mature and sexual content, offensive language, alcohol, drug, or gambling. Unfortunately, Google rarely verifies the maturity ratings provided by developers unless complains are raised. Therefore, to be certain whether or not an Android app contains harmful content, parents have to painfully test the app themselves or search for the same app in iOS App Store to verify the maturity rating. Our proposed ALM algorithm can better assist parents and children to make decisions when they choose apps on the platform of Android.

5.4 Experiment 3: Exploring Factors Contributing to Incorrect Ratings

As discussed earlier, there is a significant portion of Android apps with inaccurate maturity ratings. In this section, we

conduct some preliminary analyses to explore the factors that may lead to the inaccurate maturity ratings.

We first captured two categories of Android apps' attributes from our dataset: apps' attributes and developers' attributes. The apps' attributes included: *popularity, price, and dangerous level of the required permissions*. Developers' attributes included: *general privacy awareness, trustworthiness, actual privacy awareness, and child safety awareness*.

For apps' attributes, the number of installations was used to infer apps' popularity. Although app rank can be retrieved to represent popularity, the ranks change every day and it is difficult to keep tracking. An app's price is assigned to a binary variable: *1* if it is a paid app and *0* if it is a free app. For apps' required permissions, Chia et al [15] divided Android permissions into three categories: *danger_info* permissions, *danger* permissions, and *ok* permissions. *danger_info* permissions consist of those permitting access to users' sensitive personal information; while *danger* permissions consist of those whose actions can be harmful to users. To clarify, *danger_info* permissions are included in the set of *danger* permissions. The permissions which belong to neither *danger_info* nor *danger* categories are *ok* permissions. This experiment adopts their definitions on the three categories of permissions. Weight values of *3*, *2*, and *1* were assigned to *danger_info*, *danger*, and *ok* permissions, respectively. Therefore, for each app, the overall score for the dangerous level of all the required permissions was generated by aggregating the weights of the permissions.

For developers' attributes, the privacy regulatory culture or norm of developers' countries is used to represent developers' general privacy awareness. Smith [24] divided countries into two categories based on the privacy regulatory culture or norm: *human right* countries and *contract term* countries. According to Smith [24], *human right* countries typically have comprehensive privacy regulations which address all data collection and use within the society, whereas *contract term* countries only have regulations regarding collection and use of certain types of data, which do not extend to all types of data in all sectors of the society. Since *human right* countries have stricter privacy regulations, we argue that developers from these countries should have higher levels of privacy awareness, which should lead to higher possibility of correct maturity ratings given to their Android applications. Developers' countries are inferred from the domain of their websites, and the currency of their apps. For the countries which appeared in our dataset but were included in Smith's framework, we manually researched the privacy regulatory culture or norm of that specific country to determine its category. As a result, 18 countries were labeled as *human right* countries, while others were all categorized as *contract term* countries (as Table 8). Score values *1* and *0* are assigned to *human right* country and *contract term* country respectively to represent developers' general privacy awareness.

The distribution of developers' countries is presented in Fig. 5. The percentage of apps from *human right* countries was 25%, and the rest were all from *contract term* countries.

Table 8 List of human right countries

Human right countries
Australia, Austria, Canada, Czech Republic, Denmark, France, Finland, Germany, Hungary, Italy, Netherlands, New Zealand, Norway, Slovenia, Spain, Sweden, Switzerland, and U.K.

The evaluation of developers' websites by Web of Trust (WOT) [25] was used to represent the trustworthiness, actual privacy awareness, and child safety awareness of the developers. WOT measures websites' *trustworthiness* by testing whether websites deliver reliable services; *privacy awareness* is measured by testing whether websites keep users' personal information safe; *child safety awareness* is measured by checking whether websites only contain age-inappropriate materials. WOT obtains the evaluation of websites by wisdom-of-crowd, and it provides browser add-ons so that users can evaluate visited websites. As of November 2012, WOT has collected evaluations for over 52 million websites. Therefore, WOT's evaluations on the above three categories can reflect the attributes of Android developers' websites. For each category, WOT provides reputation scores (range from 1 to 99) for each category together with their confidence levels (range from 1 to 99). Thus, for each category, the general rating can be calculated by multiplying the reputation and confidence using the following equation:

$$rating = (reputation - 50) * confidence \qquad (7)$$

In this research, we are not only interested in finding out whether attributes of apps and developers affect the correctness of maturity ratings, but also whether the observed effects vary upon additional factors. In this experiment, three additional factors are examined: The first factor is *price*, with which we aim to observe whether the influences of apps' and developers' attributes on the maturity ratings would vary upon *paid* vs. *free* apps. The second factor is *general privacy awareness*, with which we aim to observe whether the influences of apps' and developers' attributes on the maturity ratings would vary upon *human right* vs. *contract term* countries. The last factor is *platform*, with which we aim to observe whether the influences of apps' and developers' attributes on the maturity ratings would vary upon *cross-platform apps* vs. *Android-only apps*.

Pearson's correlation and linear regression were used to conduct analysis. As shown in Table 9, *price* had negative effect on overrating, which indicates that free apps are more likely to be overrated than paid ones. We notice that the negative effect of price on overrating was found to be stronger for Android-only apps than for cross-platform apps.

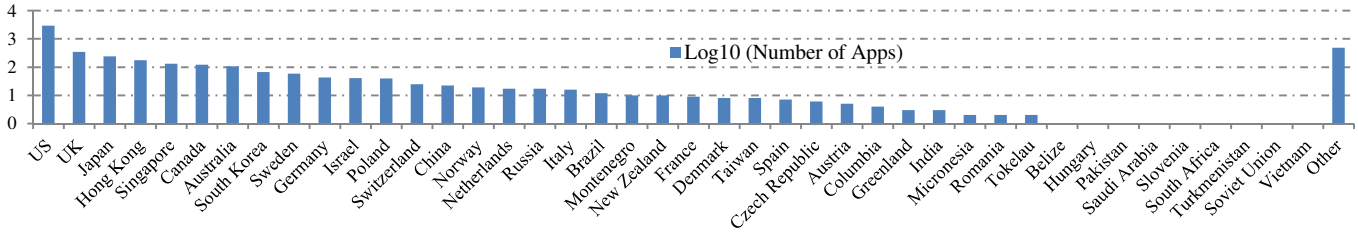

Figure 5 Distribution of developers' countries

In other words, free apps which are only available on the Android platform are more likely to be overrated. This result is interesting. Given that Android-only apps are often newly developed or published by small companies or individual developers, we argue that overrating may serve as a strategy to further attract users' attentions to those free apps. This is because, for free apps which do not require purchase to download, advertisements are the primary revenue sources. Developers are therefore more eager to promote their apps, even by giving apps the same or look-alike names as the well-known apps.

Table 9 Significant factors that affect overrating (n=265)

Impact	CP	AO	AOSL
Price→Overrating	-0.05^{***}	-0.25^{***}	-0.26^{***}
Overrating → Danger Permission	0.57^{***}	0.45^{***}	0.37^{***}

Impact	Paid	Free	HR	CT
Price→ Overrating	-	-	-0.21^{***}	-0.33^{***}
Overrating→Danger Permission	0.44^{***}	0.62^{***}	0.43^{***}	0.47^{***}

$^{***}p<0.001$ [*Note*. CP = cross-platform apps; AO = Android-only apps; AOSL = Android-only apps which have same or look-alike names with iOS apps; HR = human right countries; CT = contract term countries.]

As shown in Table 9, the negative effect of price on overrating was found to be stronger in *contract term* countries than in *human right* countries. In other words, free apps developed in *contract term* countries are more likely to be overrated than those developed in *human right* countries.

In addition, the result showed that overrating had a significant positive effect on requesting *dangerous* permissions. That is to say, overrated apps were more likely to request more *dangerous* permissions. Such effect was stronger for *cross-platform* apps than for *Android-only* apps, and stronger for *free* apps than for *paid* apps. These results suggest that those overrated cross-platform apps for free tend to be more data-hungry by requesting more data permissions across different platforms. Moreover, the positive effect of overrating on requesting *dangerous* permissions was stronger for apps developed in *contract term* countries than those developed in *human right* countries.

Table 10 Significant factors affect underrating (n=142)

Impact	CP	AO	Paid	Free	HR	CT
Trust→Underrating	**-0.26^{*}**	**-0.04^{*}**				
Privacy→ Underrating	**-0.23^{*}**	0.03				
Child Safety →Underrating	**-0.07^{*}**	-0.03				
Popularity→ Underrating			0.01	**0.10^{**}**		
Price→Underrating	-0.01	**0.10^{**}**	-	-	0.01	**0.13^{*}**

$^{*}p<0.05$, $^{**}p<0.01$, $^{***}p<0.001$ [*Note: The insignificance was not shown*.]

Data analyses were further conducted to explore factors that may lead to underrated maturity ratings. As shown in Table 10, the trustworthiness of a developer's website (evaluated by WOT) had significant negative effect on underrating. That is to say, the less trustworthy the developer's website is, the more likely that the app may be underrated. Such effect was found to be stronger for cross-platform apps than for Android-only apps.

It was found that privacy awareness level of a developer's website (evaluated by WOT) had negative association with underrating for cross-platform apps. This finding indicates that those developers with lower levels of privacy awareness are more likely to underrate the cross-platform apps, for the purpose of reaching wider user population and harvesting users' personal information.

As shown in Table 10, the child safety score of a developer's website (given by WOT) had significant negative association on underrating for cross-platform apps. That is to say, if a developer's website has a lower safety score for children, the app developed by this specific developer is more likely to be underrated.

For free apps, popularity was found to be positively associated with underrating. Such effect was not significant for paid apps. This result suggests that popular free apps are more likely to be underrated. In addition, we found that paid apps are more likely to be underrated when developers are from *contract term* countries than those from *human right* countries.

6. CONCLUSION

As discussed earlier, no research has systematically uncovered the extent and severity of these unreliable app maturity ratings, nor has any research shown the types of apps that are most often mis-categorized. To bridge this gap, our study develops mechanisms to verify the maturity ratings of mobile apps and investigates the possible reasons behind the incorrect ratings. Specifically, we develop a text-mining algorithm "Automatic Label of Maturity ratings" (ALM) to verify mobile apps' maturity ratings on the Android platform compared to Apple's iOS platform. ALM discovered that over 30% of Android apps have unreliable maturity ratings, among which 20% apps are overrated and 10% apps are underrated.

Our research has several contributions. First, we practically examine the maturity rating policies on both Android and iOS platforms, and discover the inconsistencies and ambiguities from both policies. Second, based on app descriptions and user reviews, the algorithm ALM is developed to automatically verify Android apps' maturity ratings that were based on developers' self-disclosure. Experimental results show that ALM has advanced performance on detecting objectionable content in any maturity levels in terms of precision, recall and f-score. Third, we conduct some preliminary analyses to explore the factors that may lead to untruthful maturity ratings in Android apps. We believe that our findings have important implications for platform providers (e.g., Google or Apple) as well as for regulatory bodies and application developers.

7. ACKNOWLEDGMENTS

The authors are very grateful to the anonymous reviewers for their constructive comments, and to Pamela Wisniewski for her input and editorial assistance. Part of this research was supported by the U.S. National Science Foundation under grant CAREER 0643906 and CNS-1018302. Any opinions, findings, and conclusions or recommendations expressed herein are those of the researchers and do not necessarily reflect the views of the U.S. National Science Foundation.

8. REFERENCES

[1] A. Mitchell, T. Rosenstiel, L. H. Santhanam, and L. Christian. (2012). Future of Mobile News. Available: http://www.journalism.org/analysis_report/future_mobile_news

[2] D. Hardawar. (2012). The magic moment:Smartphones now half of all U.S. mobiles. Available: http://venturebeat.com/2012/03/29/the-magic-moment-smartphones-now-half-of-all-u-s-mobiles/

[3] M. Carmichael. (2011). Stat of the Day: 25% of Toddlers Have Used a Smartphone. Available: http://adage.com/article/adagestat/25-toddlers-a-smartphone/229082/

[4] EnterpriseAppsTech. (2012). Smartphone Usage Growing Amongst Teenagers.

[5] D. Graziano. (2012). Android and iOS Still Rule the Mobile World; Microsoft and RIM Have Long Roads Ahead. Available: http://bgr.com/2012/11/02/android-ios-market-share-dominate-microsoft-rim/

[6] Statista. (2012). Number of available applications in the Google Play Store from December 2009 to September 2012. Available: http://www.statista.com/statistics/74368/number-of-available-applications-in-the-google-play-store/

[7] 148Apps.biz. (2012). App Store Metrics. Available: http://148apps.biz/app-store-metrics/

[8] Apple. (2012). App Store Review Guidelines. Available: https://developer.apple.com/appstore/guidelines.html

[9] R. H. Rasmussen. (2011). Unreliable ratings on mobile apps. Available: http://kidsandmedia.org/unreliable-ratings-on-mobile-apps/

[10] C. Kang. (2011). Inappropriate content making its way to mobile apps. Available: http://www.washingtonpost.com/business/economy/inappropriate-content-making-its-way-to-mobile-apps/2011/10/05/gIQAnYB4kL_story.html

[11] FTC, "Mobile Privacy Disclosures: Building Trust Through Transparency," 2013.

[12] FTC, "Mobile Apps for Kids: Current Privacy Disclosures are Disappointing," 2012.

[13] A. Möller, S. Diewald, L. Roalter, F. Michahelles, and M. Kranz, "Update Behavior in App Markets and Security Implications: A Case Study in Google Play," *LARGE,* p. 3, 2012.

[14] J. Lin, S. Amini, J. Hong, N. Sadeh, J. Lindqvist, and J. Zhang, "Expectation and purpose: Understanding users' mental models of mobile app privacy through crowdsourcing," in *Proceedings of the 14th ACM International Conference on Ubiquitous Computing*, 2012.

[15] P. H. Chia, Y. Yamamoto, and N. Asokan, "Is this app safe?: a large scale study on application permissions and risk signals," in *Proceedings of the 21st international conference on World Wide Web*, 2012, pp. 311-320.

[16] A. P. Felt, E. Chin, S. Hanna, D. Song, and D. Wagner, "Android permissions demystified," in *Proceedings of the 18th ACM conference on Computer and communications security*, 2011, pp. 627-638.

[17] P. G. Kelley, S. Consolvo, L. F. Cranor, J. Jung, N. Sadeh, and D. Wetherall, "A Conundrum of Permissions: Installng Applications on an Android Smartphone," in *Proceedings of the Workshop on Usable Security (USEC)*, 2012.

[18] A. P. Felt, E. Ha, S. Egelman, A. Haney, E. Chin, and D. Wagner, "Android permissions: User attention, comprehension, and behavior," in *Proceedings of the Eighth Symposium on Usable Privacy and Security*, 2012, p. 3.

[19] M. Siegler. (2009). Here's How iPhone App Store Ratings Work. Hint: They Don't. Available: http://techcrunch.com/2009/06/29/heres-how-iphone-app-store-ratings-work-hint-they-dont/

[20] T. Agten. (2012). Same Apps Have Different Age Rating in Apple App Store and Google Android Market. Available: http://www.distimo.com/blog/2012_01_same-apps-have-different-age-rating-in-apple-app-store-and-google-android-market/

[21] Apple. (2012). Application Ratings. Available: http://itunes.apple.com/WebObjects/MZStore.woa/wa/appRatings

[22] Google. (2012). Application Content Ratings. Available: http://support.google.com/googleplay/bin/answer.py?hl=en&answer=1075738

[23] NielsenWire. (2011). Play Before Work: Games Most Popular Mobile App Category in US. Available: http://blog.nielsen.com/nielsenwire/online_mobile/games-most-popular-mobile-app-category

[24] H. J. Smith, "Information privacy and its management," *MIS Quarterly Executive,* vol. 3, pp. 201-213, 2004.

[25] Web of Trust (WOT). Available: http://www.mywot.com