Identifying and Analyzing the Privacy of Apps for Kids

Minxing Liu[1]*, Haoyu Wang[1], Yao Guo[1], Jason Hong[2]
[1]Peking University
[2]Carnegie Mellon University
{liuminxing, howiepku, yaoguo}@pku.edu.cn, jasonh@cs.cmu.edu

ABSTRACT

One aspect of privacy that has not been well explored is privacy for children. We present the design and evaluation of a machine learning model for predicting whether a mobile app is designed for children, which is an important step in helping to enforce the Children's Online Privacy Protection Act (COPPA). We evaluated our model on 1,728 apps from Google Play and achieved 95% accuracy. We also applied our model on a set of nearly 1 million free apps from Google Play, and identified almost 68,000 apps for kids. We then conducted a privacy analysis of the usage of third-party libraries for each app, which can help us understand some of the app's privacy-related behaviors. We believe this list can serve as a good start point for further fine-grained privacy analysis on mobile apps for children.

Keywords

mobile applications; children's privacy; Android

1. INTRODUCTION

Mobile apps have seen widespread adoption, with over one million apps available on each of Google Play and the Apple App Store. These apps can use the rich capabilities of smartphones, including personal data (e.g., contact lists, emails, photos, and call logs) and sensor data (e.g., GPS, camera, and microphone), enabling many new kinds of user experiences and functionality. However, these same capabilities have led to many new kinds of privacy concerns and intrusions. Previous work has investigated a wide range of privacy issues with respect to mobile apps, for example finding potential leaks of sensitive information [16] or wisdom of crowds approaches to making decisions about sharing data [7, 22, 21]. However, one area that is relatively unexplored is privacy for children.

*Most of this work was done when Minxing was a visiting student at CMU.

HotMobile '16, February 26-27, 2016, St. Augustine, FL, USA

DOI: http://dx.doi.org/10.1145/2873587.2873597

According to a 2013 survey, 75% of children under age 8 are using mobile devices [18]. In the United States, privacy protection for children is especially important due to the Children's Online Privacy Protection Act [1], or COPPA. Passed in 1998, COPPA regulates actions of operators of online services (thus including mobile apps) that are targeted at children under age 13. COPPA requires operators to only collect necessary information from children, offer a clear description of what information will be collected and for what purpose, and obtain consent from parents. Furthermore, any collected information should not be made publicly available in an identifiable form.

Currently, the task of enforcing COPPA falls mainly to the Federal Trade Commission (FTC), which has levied several warnings and fines for violations [3]. The FTC has also published two reports examining app privacy in the context of children in 2012 [13, 14]. In the first report, the FTC manually checked 200 apps from Google Play and the Apple App Store respectively, and found that there was little or no information available to parents about the privacy of apps on the product detail pages[1]. In a follow-up report six months later, the FTC found little improvement to privacy information. The FTC also downloaded and tested apps and found that many apps shared kids' information with third parties without disclosing these practices to parents.

These two FTC reports offer insights into some of the privacy issues with respect to smartphone apps for kids. However, a major limitation is that, today, there is no automated way of identifying and analyzing the privacy-related behaviors of apps that target children. To underscore this point, the FTC used a manual and highly labor-intensive process for their reports. First, FTC employees gathered a set of apps by searching for "kids". Second, FTC employees examined each app's description to identify whether an app actually targets kids or not. The reports found that about 25% of apps collected were actually not directed at children [13]. Third, FTC employees manually downloaded and checked if a given app linked to any social media, allowed in-app purchases, or used advertising. Fourth, FTC employees intercepted Internet traffic for a given app to see if it transmitted device ID, phone number, and geolocation information.

While the FTC's approach was fairly comprehensive, the amount of labor involved severely limits how many apps can be inspected, and how often. For example, the FTC was

[1]The webpage that includes all detailed information of an app, e.g., description, icon, screenshots, etc.

able to only inspect 400 apps, and has not repeated their work since 2012. As such, our long-term goal is to improve the speed, accuracy and scalability of inspecting apps for kids by introducing a series of automated methods. In this paper, we present some of our initial results.

A general pipeline for identifying potentially problematic apps can be derived from the method used by the FTC: (1) gather a set of apps, (2) identify which apps likely target children, (3) analyze the apps to see which ones use social media, in-app purchases, or advertising, and (4) apply static and dynamic analysis techniques to see which apps transmit potentially sensitive data. The work presented in this paper focuses on the first three steps. More specifically, we developed a machine learning classifier that uses several text-based and image-based features to identify apps designed for kids, which achieved 95% accuracy. We also ran our classifier on almost 1 million apps from Google Play, and for each app classified as being for kids, we retrieved a privacy grade from *privacygrade.org* [5] and applied simple static analysis techniques to see if it used social media, in-app purchases, or advertising.

Our work has at least three potential applications. The first is to help regulators like the FTC and their equivalent in other countries. Regulators can use our tool to get a list of potentially problematic apps, which can then be used to prioritize which apps they will manually inspect further (e.g. the most popular and problematic apps for kids). Given that regulators often have limited resources, this approach can help them in making their work more comprehensive and ongoing. The second is to help parents. For example, third parties, such as *Consumer Reports* or *privacygrade.org*, could make the results of these analysis easily browsable and searchable by the general public, making it easy for parents to understand what potential problems there might be before downloading an app. The third is to help app store administrators, who can use the information to better label which apps are for kids, help flag apps for further inspection, or nudge developers to be aware of potential legal issues they may be violating when they upload apps.

This paper makes the following research contributions:

- We present the design of a machine learning classifier that can identify whether a mobile app is directed towards children. Our classifier uses both text-based and image-based features extracted from an app's product detail page. To the best of our knowledge, we are the first to build a classifier to recognize mobile apps for kids.
- We evaluate our approach on 1,728 apps. The results of 10-fold cross validation yield an accuracy of about 95% and over 91% for both precision and recall. We also discuss which features are the most effective in predicting whether an app is directed to kids.
- We also apply our classifier on nearly 1 million apps, and analyze the privacy-related behaviors of apps targeting kids. Our results can serve as a good start point for further fine-grained analysis.

2. RELATED WORK

There are two major lines of related work. The first are studies that use data mining to analyze mobile app markets [17, 15, 26, 10]. Examples include analyzing correlations between apps' technical, customer, and business aspects, e.g. relation between download volume and price [17], detecting ranking fraud [26], or detecting similar apps [10]. Some past work here also looked at privacy of apps, though not in the context of children. For example, WHYPER [23] leveraged natural language processing (NLP) techniques to infer the usage of sensitive data from app descriptions. They then detected the inconsistencies between the description and real app behaviors related to fetching private data to determine the privacy performance of an app. Wang *et al.* [24] utilized text-mining skills on the package/class/variable names used in the custom code to infer the purpose of privacy-related behaviors of an app.

Another line of work focuses on potential risks of online services directed to children. As mentioned above, the FTC released two reports inspecting privacy issues in mobile apps designed for children in 2012 [13, 14]. Liccardi *et al.* [20] developed a new framework to help mobile app developers comply with COPPA. Specifically, they provided a simple and efficient interface for developers to state their usage of personal data so that parents could understand potential privacy risks. Chen *et al.* [12] and Bhoraskar *et al.* [8] both focus on in-app advertisements of "kids' apps" and check if they include inappropriate content for children or if they attempt to collect personal information.

The closest research is Chen *et al.* [11] and Hu *et al.* [19], which focus on unreliable content rating of apps. They proposed algorithms to detect mature content in an app and assigned it an accurate maturity level. Our work differs as the content rating of an app does not necessarily denote its intended users. For example, while apps with "High Maturity" are usually not designed for children, apps with content rating "Everyone" does not mean it is designed for children. Thus, their work might assist us in improving the accuracy, but we focus on a different goal.

3. IDENTIFYING APPS FOR KIDS

3.1 Overview

In this section, we present the design of our machine learning classifier for identifying mobile apps directed to children. The classifier accepts a feature vector based on content extracted from an app's product detail page. We use manually labeled data from Google Play to train the classifier.

3.2 Feature Extraction

We want features that are relatively simple and fast to calculate (to help with scalability), and general enough for different app markets (e.g. Apple App Store). After manually examining several popular apps for kids, we chose 171 features, summarized in Table 1 and described below.

3.2.1 Meta Features

This set of features describes basic metadata about an app. Here we extracted 2 features. The first is **Category**, which is a binary value indicating whether or not the app belongs to a common category for "kids' apps", namely *Education, Games, Comics, and Entertainment.* Intuitively, this feature should be effective in identifying apps for kids, but should also have many false positives since many apps in these categories also target adults.

Table 1: The features used in our classification model.

Category	Feature Description	Details
App Category	Category of an app	A binary value, which represents whether the app belongs to a relevant category where most "kids' apps" are classified (Education, Games, Comics, or Entertainment).
Content Rating	Content rating of an app	An ordinal value, which represents corresponding content rating of the app.
Title	Frequency and importance of key words from the title	A 5 dimension vector, each value representing the TF-IDF value of five key words, namely "children", "fun", "game", "kid" and "toddler".
Description	Frequency and importance of key words extracted from app description	A 10 dimension vector, each value representing the TF-IDF value of ten key words, namely "animal", "children", "education", "fun", "game", "kid", "learn", "play", "preschool" and "toddler".
Readability of the description	Readability score	A value that represents the readability of the description using the Flesch-Kincaid readability test [2].
Picture Resources	Color distribution and usage of the icon and screenshots	A 49 dimension vector for each picture resource, and a total of 147 (49×3) dimensions (icon+two screenshots). Features of a picture resource include the color histogram, average hue, average saturation, average brightness value, and number of colors used.
Strings on Screenshots	Frequency and importance of key words extracted from strings in screenshots	A 6 dimension vector. The first five represent TF-IDF values of key words selected, specifically "baby", "children", "fun", "kid" and "play". The last one represents the length of strings on the screenshots.

The second is **Content Rating**. There are five different content ratings on Google Play: *Everyone, Low Maturity, Medium Maturity, High Maturity* and *Unrated*. Usually, apps for kids are tagged with *Everyone* or *Low Maturity*. Similar to the category feature, content ratings should identify many "kids' apps" but also have many false positives. For example, many calculator apps are rated "Everyone". We mapped the 5 different ratings to a corresponding numerical value (1-5).

3.2.2 Title and Description

This set of features focuses on the text describing an app. For both title and description, we first split the text into a bag of words, filtering out non-ascii words, punctuation, and stop words like "the". Then we used the Porter stemming algorithm [4] to identify the root of a word, combining singular forms and plural forms of words, such as "kid" and "kids". Next, we calculated a TF-IDF value for each word to denote its importance. Common words within a document (Term Frequency) but relatively rare in other documents (Inverse Document Frequency) will have high scores. Each value is in the range of [0.0, 1.0]. To calculate TF, we counted the number of times each word occurs in a given text. To calculate IDF, we used our entire training corpus, which contains 1,728 labeled apps from Google Play. Note that we completed the process above separately when extracting features from the title and from the description.

To optimize, we reduced the number of dimensions using the Chi-square test to select words that are the most efficient for classification, and only calculating TF-IDF values for these key words as features. This technique was first proposed by Yang *et al.* [25], who found that preserving just the most representative key words will generally obtain similar or even improved average accuracy. The key words for titles and descriptions are described in Table 1.

3.2.3 Readability of App Description

Intuitively, apps for kids are more likely to have descriptions that are easier to read and understand. To represent this feature, we ran the Flesch-Kincaid readability test [2] on the description and generate a score based on the total number of sentences, words, and syllables (with lower scores meaning easier to read).

3.2.4 Color Features from Icon and Screenshots

In our initial investigations, we found that the picture resources of "kids' apps" often use bright primary colors with colorful backgrounds. In contrast, many apps not targeting kids tend to use colors from a wider palette of colors, and few seem to use highly saturated colors.

To extract color-related information, we represented each picture in hue, saturation, and brightness value (HSV) format. Concretely, we calculated 49 features for the app icon and associated screenshots. Google Play requires each app to upload at least two screenshots. We only use the first two screenshots if there are more than two. Thus we have a total of 147 (49×3) features.

We have three features representing the average values for each of hue, saturation, and brightness value, and one more for the total number of colors used. The remaining features represent the HSV histogram. There is a tradeoff here between fidelity of the histogram and dimensionality. A fine-grained histogram might better capture the distribution of colors, but will also lead to a high-dimensional set of features with relatively sparse data. After trying a variety of groupings on a small set of apps, we chose to represent the histogram with a granularity of 3 hues, 3 saturations, and 5 brightness values, leading to 3×3×5 or 45 "color groups". We then calculated the relative proportion of each color group for the icon and screenshots, each ranging between [0.0, 1.0].

3.2.5 Text in Screenshots

We also extracted text from screenshots using Tesseract-OCR [6], an open source OCR library. We conjectured that the words used in an app could help with identifying if it were for kids. We extracted text from all screenshots and aggregated them into a single string. Then, similar to the processing for title and description, we applied TF-IDF and used the top 5 key words as features.

We also used the average length of the string in all screenshots as a feature. In our initial explorations, we found that the length tends to be shorter for "kids' apps". A possible explanation is that these apps have fewer words. An alternative is that "kids' apps" tend to use colorful and exaggerated fonts that the OCR library fails to recognize.

3.2.6 Features from APK File

We also extracted features from the Android Application

Package (APK) file, including strings and picture resources stored in the APK file. We used similar text and picture analysis techniques to extract the features. However, in practice, we found that these features offered marginal improvement to the overall accuracy. Therefore, we chose not to integrate APK level features in our current classifier and the details will be omitted in this paper.

4. EVALUATION

4.1 Data Collection

To gather our data set, we chose to use sites that reviewed apps for kids, so as to minimize subjective judgement on our part in terms of what keywords to search for on Google Play, as well as identifying which apps are targeted at children. More concretely, we selected a set of key words to search for on the Google search engine, e.g. "Android apps for kids" or "Android apps for preschool". The key words we used came from the FTC report [13], all of which are variants of the word "children". Among the search results (using just the first three pages to ensure relevance), we looked for review websites and recommendation lists, which often had titles like "The best Android apps for your kids". Finally, we downloaded each app mentioned in those lists and eliminated duplicate apps and apps not available on Google Play. Using this method, we downloaded 576 apps directed towards children, or "positive examples".

We also collected apps not targeting children as counter-examples. We collected 12 key words such as "men", "women", "college students", and other key words from the FTC report. Using the same method as above, we downloaded a total of 804 apps. We also used a list of key words from the same FTC report describing categories of apps, such as "Educational", "Game", "Animal-related", and "Math". We then combined them with the previous 12 key words and made new search queries, e.g., "Android math apps for college students" and used the same method to collect 348 more apps. This approach gave us a wide range of apps and should help prevent overfitting. For example, we do not want our classifier to simply predict an app to be for kids because it contains the word "Math" in its title. The total number of negative examples in our dataset is 1,152 (804+348).

4.2 Evaluation Method

We first normalized the value range of our features to [0.0, 1.0], and then used LibSVM [9] to train our classifier. We did a grid search of models and parameters, and chose the *radial basis function* kernel and best parameters (cost=1.0, gamma=0.125, degree=3) for our classifier. We then used 10-fold cross validation to test the performance of our classifier. Concretely, we measured true positives (TP), false positives (FP), true negatives (TN), and false negatives(FN), and then calculated accuracy, precision, recall, F-measure, and Area under the ROC curve.

4.3 Results and Analysis.

Overall Result. Our classifier achieved an overall accuracy of 95%, with 93% precision and 91% recall. Table 2 presents detailed results of different evaluation metrics.

Compared with baselines. Table 3 compares our results with four selected baselines. Overall, our classifier has better performance than these baselines.

Table 2: Performance Overview.

Precision	Recall	F-measure	AUC	Accuracy
0.933	0.915	0.924	0.983	94.97%

Table 3: Comparison of our classifier to baselines.

Baseline description	Precision	Recall
Use app category as the only feature	0.624	0.945
Use content rating as the only feature	0.460	0.815
FTC search on app market	0.757	N/A
"Family" tag of Google Play	0.988	0.436

The first baseline uses the App Category. Specifically, we removed all the other features and trained a new model using the same algorithm (LibSVM). It achieves good recall because most apps for kids fall into the Education or Games category. However, these categories also contain many apps not targeting kids, e.g., educational apps for college students or adults, thus leading to relatively low precision.

The second baseline uses Content Rating and its precision and recall are both significantly lower than ours.

Then we compared against the search results of Google Play, referring to the results from the FTC report [13]. This is not an actual algorithm, but we compare their precision results against ours. The FTC staff searched for "kid", and after manual checks found that 24.25% of apps did not actually target kids, but rather parents or teachers[2]. Compared to them, only 7% of apps predicted as "kids app" by us are actually not targeted at kids.

For our fourth baseline, we used Google Play's "Family" category[3], which is a special category that draws on apps from other categories. It even has subcategories specifying different age groups of kids, like "kids 6-8". We looked at our training data and found that only 254 apps in our data set were in the Family category, with 251 positive examples and 3 negative ones (which targeted parents). It seems that the Family category is done manually, and so can achieve high precision but lacks scalability, causing low recall.

Error Analysis. Here, we examine misclassified apps, including false positives (apps incorrectly classified as "kids app") and false negatives (apps targeted at kids not recognized).

For false positives, most are borderline cases, typically games for both kids and adults but not specifically targeting kids. These apps often include colorful pictures to attract users. Some games, e.g., "*ca.samsstuff.samstictactoe*", also include words like "children" in its description. As such, the features extracted from these apps resemble apps specifically targeting kids.

For false negatives, most cases come from the apps whose title and description do not include common keywords. For example, "*kr.co.smartstudy.cartown_android_googlemarket*" is an app where you can "sing, drive and play with your favorite cars," but does not have any of our keywords as listed in Table 1. This finding suggests a need for improving the analysis for Title and Description.

[2]The report claimed that some of these apps are also for kids too. In our paper, we aim to find apps that are designed primarily for kids. Thus, we regard all of these apps as wrongly classified.

[3]The special Family category was created from June 2015.

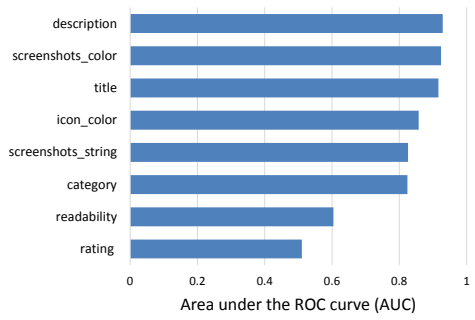

Figure 1: Area under the ROC curve (AUC) of different features used by our classifier.

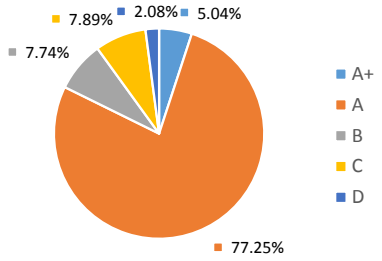

Figure 2: Distribution of Privacy Grades

Most Important Features. Figure 1 shows the relative importance of features using Area Under the ROC Curve (AUC), with higher AUC values being better. Some features stand out, including *text-based information from title and description*, and *color-related information from icon and screenshots*. Apart from the title and the description, features from image resources are also effective. Another image-based feature, OCR strings from screenshots, works surprisingly well at about 0.83 AUC.

As for other features, "Category" is a good indicator. Contrary to our expectations, "Readability" is not quite effective, perhaps because developers typically write for parents and also it is dependent on developers' own writing style. "Rating" is not effective either, partly because the inefficient rating system of Google Play [11, 19].

5. PRIVACY ANALYSIS

We applied our classifier on a large set of free apps from Google Play and conducted a preliminary privacy analysis. The use case is to find the most egregious apps in the context of COPPA, which can help regulators focus their resources on ones that should be further investigated manually.

List Generation. We collected 977,948 free apps from Google Play ending around April 2015, including their meta information, titles, descriptions and picture resources. For each app, we extracted features as described in Section 3 and fed them into our classifier. Our classifier identified a total of 67,778 apps targeting kids (∼6.9% of the data set).

Privacy Analysis. For each identified "kids' app", we analyzed three potential privacy issues. First, we examined its privacy grade, as retrieved from *privacygrade.org* [5]. Each app is assigned a privacy grade, one of A+, A, B, C and D, as calculated by a machine learning model trained on labeled training data collected from crowdsourcing. More details can be found in previous work [21, 22].

Second, we examined three behaviors of interest to the FTC [13], namely (1) whether the app uses targeted advertising, (2) whether the app can connect with a social network (e.g., has "share to Facebook" option), and (3) whether the app offers in-app purchases. For targeted advertising and social network, we compared the package names of libraries in the app with a list of third-party libraries known to be relevant with targeted ads or social networking, using results previously compiled by *privacygrade.org*. For in-app purchases, this feature must be declared in the app's manifest file. We simply decompiled each app and looked for that. Note that just because the app contains a given library or permission does not mean that it actually uses it. However, for our given use case, we can expect a person to manually inspect the app to verify.

Table 4: Usage of interactive behaviors in the app.

Category	Targeted Ads	Social network	In-app purchase
Percentage	53.0%	19.6%	22.5%

Third, we examined the app's popularity. We used the download volume as the popularity of an app, which can be found on its detail page. Generally, an app with more downloads should be paid more attention.

We generated a table of 67,778 rows with the above information. The distribution of privacy grades of these apps is shown in Figure 2. About 82% of apps get an A or A+, which means that these apps use few permissions for unusual purposes. In contrast, about 10% get a C or D, which could warrant more attention.

Table 4 shows the usage of three interactive behaviors discussed above. About 53% of the apps include targeted ads. This high percentage can partly be attributed to the fact that we have only analyzed free apps, which typically use ads to make money. About 20% of the apps use social networks, and another 22% have in-app purchases. Again, these behaviors alone are not necessarily violations of COPPA, but may suggest further attention.

6. DISCUSSION

6.1 Limitations

Incomplete or wrong information on app markets. Currently, we extract features from the detail page of Google Play for each app. We selected features that should also be available in other app markets, e.g., description, icon, screenshots, etc. However, our approach assumes that developers do not provide misleading information, which would clearly impact our approach. As such, it may be helpful to extract features from the APK file, since it represents the actual behavior of each Android app. Although adding APK features cannot improve the current results, they might be used as an alternative or as a complementary approach if developers start to deliberately put misleading data on app markets.

Data collection. Our data set is based on review websites. This approach minimizes certain kinds of biases but may introduce others. For example, most review sites do not look at the long tail of apps.

App coverage. Currently, our work focuses on apps that specifically target kids, but the FTC has also fined

other kinds of apps that collect data from children under the age of 13. For example, neither Yelp nor Path specifically target children, but they were fined by the FTC since they explicitly asked users for their age and still collected data from people who stated they were under the age of 13. Our approach does not address this problem.

Future privacy analysis. We presented an initial privacy analysis in Section 5 using fairly general approaches. For future work, we would like to develop automated techniques to address issues more specific to children and their parents. For example, COPPA requires app developers to provide notice on the product detail page about what information will be collected from children. Program analysis can be used to identify what information will be collected by the app at runtime. These results can then be cross-checked using NLP techniques on app descriptions or terms & conditions page. As another example, we can develop algorithms to detect if parents are involved at any point in data collection, e.g. looking for certain kinds of dialog boxes.

6.2 Further Implications for Privacy

In general, privacy for kids is less ambiguous and contentious than privacy for adults, given widespread agreement that children are a vulnerable population, the detailed laws, and clear enforcement mechanisms by regulators. While improving privacy for children is a useful goal in itself, it might also be a potentially powerful leverage point for advancing privacy for all people in general. For example, some developers might not want to collect certain kinds of personal information due to the challenge of identifying children and the increased risk of enforcement. As another example, app stores might compel developers to do better with respect to privacy for kids before uploading their apps, which could in turn help educate developers about other best practices for privacy.

7. CONCLUSION

We presented the design and evaluation of a classifier to predict whether an app is designed primarily for kids. We extracted several features from the detail page of an app and evaluated the classifier on a set of 1,728 labeled apps, achieving an accuracy of 95%. We also ran our classifier on a large set of apps to generate a list of apps for children and conducted some privacy analysis on them. Our method and results can benefit regulators, parents, third-parties, and app stores in understanding and improving privacy.

8. ACKNOWLEDGMENTS

This paper is supported in part by National Science Foundation (CNS1228813), the Air Force Research Laboratory (FA8750-15-2-0281), the National Basic Research Program of China (973) under Grant No. 2015CB352201, and the National Natural Science Foundation of China under Grant No. 61421091, 61103026.

9. REFERENCES

[1] COPPA - Children's Online Privacy Protection Act. http://www.coppa.org/coppa.htm.

[2] Flesch-Kincaid readability test. https://en.wikipedia.org/wiki/Flesch%E2%80%93Kincaid_readability_tests.

[3] FTC's first fines for violating online kids' privacy law. http://www.computerworld.com/article/2592253/government-it/ftc-assesses-first-fines-for-violating-online-kids–privacy-law.html.

[4] The porter stemming algorithm. http://tartarus.org/martin/PorterStemmer/.

[5] Privacygrade: Grading the privacy of smartphone apps. http://privacygrade.org/.

[6] Tesseract-ocr. https://github.com/tesseract-ocr.

[7] Y. Agarwal and M. Hall. ProtectMyPrivacy: detecting and mitigating privacy leaks on iOS devices using crowdsourcing. In *MobiSys*, 2013.

[8] R. Bhoraskar, S. Han, J. Jeon, T. Azim, S. Chen, J. Jung, S. Nath, R. Wang, and D. Wetherall. Brahmastra: Driving apps to test the security of third-party components. In *USENIX Security Symposium*, 2014.

[9] C.-C. Chang and C.-J. Lin. LIBSVM: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology*, 2011. Software available at http://www.csie.ntu.edu.tw/%7Ecjlin/libsvm.

[10] N. Chen, S. C. Hoi, S. Li, and X. Xiao. SimApp: A framework for detecting similar mobile applications by online kernel learning. In *WSDM*, 2015.

[11] Y. Chen, H. Xu, Y. Zhou, and S. Zhu. Is this app safe for children?: A comparison study of maturity ratings on Android and iOS applications. In *WWW*, 2013.

[12] Y. Chen, S. Zhu, H. Xu, and Y. Zhou. Children's exposure to mobile in-app advertising: An analysis of content appropriateness. In *SocialCom*, 2013.

[13] F. T. Commission et al. Mobile apps for kids: current privacy disclosures are disappointing, 2012.

[14] F. T. Commission et al. Mobile apps for kids: Disclosures still not making the grade, 2012.

[15] B. Fu, J. Lin, L. Li, C. Faloutsos, J. Hong, and N. Sadeh. Why people hate your app: Making sense of user feedback in a mobile app store. In *KDD*, 2013.

[16] C. Gibler, J. Crussell, J. Erickson, and H. Chen. Androidleaks: Automatically detecting potential privacy leaks in Android applications on a large scale. In *TRUST*, 2012.

[17] M. Harman, Y. Jia, and Y. Zhang. App store mining and analysis: MSR for app stores. In *MSR*, 2012.

[18] D. Holloway, L. Green, and S. Livingstone. Zero to eight: Young children and their internet use. *LSE London, EU Kids Online*, 2013.

[19] B. Hu, B. Liu, N. Z. Gong, D. Kong, and H. Jin. Protecting your children from inappropriate content in mobile apps: An automatic maturity rating framework. In *CIKM*, 2015.

[20] I. Liccardi, M. Bulger, H. Abelson, D. Weitzner, and W. Mackay. Can apps play by the COPPA rules? In *PST*, 2014.

[21] J. Lin, S. Amini, J. I. Hong, N. Sadeh, J. Lindqvist, and J. Zhang. Expectation and purpose: Understanding users' mental models of mobile app privacy through crowdsourcing. In *UbiComp*, 2012.

[22] J. Lin, B. Liu, N. Sadeh, and J. I. Hong. Modeling users' mobile app privacy preferences: Restoring usability in a sea of permission settings. In *SOUPS*, 2014.

[23] R. Pandita, X. Xiao, W. Yang, W. Enck, and T. Xie. WHYPER: Towards automating risk assessment of mobile applications. In *USENIX Security Symposium*, 2013.

[24] H. Wang, J. I. Hong, and Y. Guo. Using text mining to infer the purpose of permission use in mobile apps. In *UbiComp*, 2015.

[25] Y. Yang and J. O. Pedersen. A comparative study on feature selection in text categorization. In *ICML*, 1997.

[26] H. Zhu, H. Xiong, Y. Ge, and E. Chen. Ranking fraud detection for mobile apps: A holistic view. In *CIKM*, 2013.