iROGUE: Identifying Rogue Behavior from App Reviews

Vaibhav Garg, Hui Guo, Nirav Ajmeri, Saikath Bhattarcharya, Munindar P. Singh
vgarg3@ncsu.edu, hguo@quora.com, nirav.ajmeri.bristol.ac.uk, saikath.bhattacharya@gmail.com,
mpsingh@ncsu.edu

***Abstract*—An app user can access information of other users or third parties. We define rogue mobile apps as those that enable a user (abuser) to access information of another user or third party (victim), in a way that violates the victim's privacy expectations. Such apps are dual-use and their identification is nontrivial. We propose iROGUE, an approach for identifying rogue apps based on their reviews, posted by victims, abusers, and others. iROGUE involves training on deep learning features extracted from their 1,884 manually labeled reviews. iROGUE first identifies how alarming a review is with respect to rogue behavior and, second, generates a rogue score for an app. iROGUE predicts 100 rogue apps from a seed dataset curated following a previous study. Also, iROGUE examines apps in other datasets of scraped reviews, and predicts an additional 139 rogue apps. On labeled ground truth, iROGUE achieves the highest recall, and outperforms baseline approaches that leverage app descriptions and reviews. A qualitative analysis of alarming reviews reveals rogue functionalities. App users, platforms, and developers should be aware of such apps and their functionalities and take measures to curb privacy risk.**

I. INTRODUCTION

With the expansion of mobile technologies, privacy threats arise not only from malicious or careless app developers, but also from app users. The privacy expectations of an app user or a third party (i.e., a *victim*) are violated when the victim (1) doesn't know about another user (i.e., an *abuser*) accessing the victim's information (spying) or (2) may know about the access, but is uncomfortable with it. The latter case includes incidents of forced consent or when public information on apps (such as a dating platforms) is accessed beyond the victim's level of comfort, such as profile stalking. We use term *rogue behavior* to mean these two types of information access, and use *rogue apps* to mean the apps that enable rogue behavior.

Research (Chatterjee et al., 2018; Freed et al., 2019; Havron et al., 2019) shows that rogue apps may cause discomfort, fear, and potential harm to the victim. Possible ways to prevent this risk include highlighting these apps and their rogue functionalities to users, warning app distribution platforms, and informing app developers. All these actions rely upon identifying rogue apps and their functionalities. Our proposed approach, iROGUE, shows how to do so.

Previous studies (Chatterjee et al., 2018; Roundy et al., 2020) focus only on the access performed without the victim's knowledge, but do not consider cases when the victim is uncomfortable of the access (of public information) even if aware of it. Chatterjee et al. Chatterjee et al. (2018) use app descriptions to identify intimate partner surveillance (IPS) apps (subset of rogue apps), the apps that someone can use to spy on his or her intimate partner (spouse, boyfriend, or girlfriend). Such IPS apps are dual-use apps, that have a legitimate purpose but are misused for spying. This concept of dual-use apps also applies to the general setting of rogue apps. Since app descriptions of rogue apps indicate only intended legitimate behavior, app descriptions may not be suitable for identify all misuses. We leverage app reviews to identify all misuses of such apps. We observe that the reviews of an app describe rogue functionalities, its misuse (potential and actual), and the privacy expectations of users. Such reviews are evidence of rogue behavior and should be brought to the attention of users, developers, and app platforms.

Example 1 shows three reviews (edited for grammar), taken from the Apple's App Store Appstore, and are relevant to the rogue behavior. Although our study is based on Apple App Store's reviews, iROGUE can be applied on reviews from other sources, including Google's Play Store Playstore.

In Example 1, the first review for AirBeam Video AirBeam, addresses the scenario where the app assists a user to access a victim's information without the victim's knowledge. AirBeam Video is a surveillance app to be installed on the abuser's device. Hence, the victim may not be an app user but a third party. The second review, for Life360 Life360, complains about the problem of inappropriate access of user's location by the user's mother. Due to the unequal power dynamics between the victim (reviewer in this case) and the abuser (mother in this case), the victim is forced to install apps that violate privacy. The third review, from 3Fun Threefun, describes the story of improper access of profile pictures. Even though the profile pictures are public, the victim is uncomfortable with the access. It is common for users to upload such information (pictures in this case) on an app. When doing so, they hold expectations of how other users would access it. Information access, as shown in these three, cases may lead to discomfort, fear, or potential harm (Freed et al., 2019; Havron et al., 2019). Thus, despite such cases of information access being common, they should be brought in front of app developers and platforms. However, app descriptions don't reveal possibility of a user (victim) to be uncomfortable of such access.

To address victims' privacy expectations, we propose the following research questions:

$RQ_{identify}$. How can we identify rogue apps from reviews?
$RQ_{functionality}$. How can we uncover rogue functionalities?

Example 1: Cases Relevant to Rogue Behavior

Fly on the wall!
(for the AirBeam Video Surveillance app AirBeam)
"with this app, i can spy on my family without them knowing it! it's such an awesome app!"

This app basically ruined my family to an extent
(for the Life360 app Life360)
"My mother made everyone in the family get this app. She freaks out when the app doesn't do its job because of random obstacles that mess with the location accuracy. Drains the battery and makes my parents paranoid to know where I am at all times. I don't even do any bad stuff, yet years of trust building are being swept away by the ability to spy on the children of a household. If you're a parent I highly recommend you don't get this app because it is extremely uncomfortable to have and it makes parents trust their children less."

Honest
(for the 3Fun: Threesome & Swingers app Threefun)
"...A lot of the local people I've talked to (Male half of a couple) have been guys who are saying they're part of a couple, and in all reality are single guys just looking to collect pictures. There is no way to report that that is why you are reporting them. It's just a boilerplate report feature. I feel there should be a way for the 3Fun community to point out people for bad behavior like this."

Section III To address $\mathbf{RQ_{identify}}$, we propose iROGUE, an approach that is trained on the deep learning features extracted from 1,884 app reviews. iROGUE includes three phases (described in Section III). First, it assigns an *alarmingness* score to each review. The alarmingness score is used to rate and rank each review according to the claims and severity of rogue behavior. Second, iROGUE identifies rogue apps, based on a *rogue score*, computed by aggregating the alarmingness scores of an app's reviews. The rogue score ranks each identified app, according to the rogue behavior reported in app reviews. Such ranking can be useful for app distribution platforms, such as Apple App Store Appstore and Google Play Store Playstore, to prioritize the scrutiny of identified apps. Third, iROGUE identifies additional rogue apps, by examining apps in the other datasets. To evaluate the performance of iROGUE, we report its precision, recall, and F1 score in identifying rogue apps.

To address $\mathbf{RQ_{functionality}}$, we leverage reviews with the top 10 alarmingness scores and manually analyze them to find their rogue functionalities. We further installed a few rogue apps on an iOS device to verify reported rogue functionalities and include our findings in Section IV. We contacted Apple App Store Appstore. We shared with them the list of identified rogue apps, along with reviews containing evidence against each app. They told us they will investigate the rogue apps and reach out to developers to rectify apps.

Contributions. Our work's novelty lies in leveraging app reviews to identify rogue apps and their rogue functionalities. We introduce assigning alarmingness scores to reviews and rogue scores to apps, based on the reported rogue behavior. We contribute to mobile app security by providing:

- iROGUE, an app reviews based approach for identifying rogue apps and their functionalities.
- A ranked list of rogue apps along with their alarming reviews revealing rogue behavior.

Organization. The rest of this paper is organized as follows. Section II describes our preliminary investigation that shows that app reviews contain evidence of rogue behavior. Section III describes our proposed iROGUE approach to identify rogue apps, along with its evaluation. Section IV shows the procedure to uncover rogue functionalities of rogue apps. Section V lists related work on information access in mobile apps. Section VI concludes this paper.

II. APP REVIEWS REVEAL ROGUE BEHAVIOR

We now describe rogue behavior reported in app reviews.

A. Seed Dataset

Chatterjee et al. Chatterjee et al. (2018) identify 2,707 iOS apps as potentially IPS. Out of these apps, they confirm 414 apps to be IPS, using semi-supervised pruning.

When we collected our data, 724 of Chatterjee et al.'s 2,707 apps (including 125 IPS) were already removed from the Apple App Store Appstore, meaning only 1,983 were available. Of these 1,983 apps, 1,687 received at least one review from 2008-07-10 to 2020-01-30, yielding 11.57 million reviews in all. Only 210 of these 1,687 apps were on Chatterjee et al.'s IPS list. Table I describes our *seed dataset*, which comprises these 1,687 apps and their 11.57 million reviews.

TABLE I
DETAILS OF OUR SEED DATASET.

App Type	Apps	Apps w/ Reviews	Reviews
Removed	724	–	–
IPS apps	289	210	190,584
Other apps	1,694	1,477	11,381,377
Total	2,707	1,687	11,571,961

B. Investigating Reviews

Since the seed dataset contains 11.57 million reviews, it is impractical to manually check each review for rogue behavior. Hence, we sampled app reviews containing at least one keyword related to rogue behavior. To form a set of such keywords, we initialized a set with the words: *spy*, *stalk*, and *stealth*. We queried WordNet Miller (1995) for synonyms of these words. We performed the query operation until we didn't find any new word in the set. The resulting set contained keywords: *spy*, *stalk*, *stealth*, *descry*, *chaff*, and *haunt*. However, we didn't consider *chaff* and *haunt* to be relevant for describing rogue behavior in reviews. Also, *descry* is present in only two reviews, both of which are

irrelevant for rogue behavior. To expand the set of keywords, we explored other corpora such as PyDictionary Pydictionary and Thesaurus Pythesaurus but did not find synonyms that are widely used in app reviews. Moreover, keywords used in the previous study Chatterjee et al. (2018), such as *track* and *control* bring many false positive reviews. For example, "I like tracking my distance when I walk with my dog." and "...you can also control the audio of your mac through the app ...I can control music tracks without having to touch the computer." are not relevant. Thus, our relevant set of keywords reverts to *spy*, *stalk*, and *stealth*. Table II shows the occurrence of each keyword, in reviews of the seed dataset. We refer to this set as *our keywords*.

There are 5,287 reviews containing at least one of our keywords. From these 5,287 reviews, we randomly sampled 995 reviews for manual scrutiny. This sample involves 179 apps with between 1 and 237 reviews each.

TABLE II
OCCURRENCE OF EACH KEYWORD.

Keyword	Review Count
Spy	2,479
Stalk	2,605
Stealth	218
Total Unique	5,287

The first author manually checked 995 reviews for rogue behavior. Out of 995 reviews, we found 402 reviews (of 83 apps in this sample) reporting rogue behavior. Our manual analysis categorize these 402 reviews along the dimensions of *story* and *reviewer*. Based on rogue story, we observe reviews of following two types:

***Rogue Act*:** Reviews describing someone performing a rogue behavior. In such reviews, the reviewer is sure about the app's rogue functionality.

***Rogue Potential*:** Reviews express the possibility of rogue behavior. The reviewer may not be sure of rogue functionality, but identifies risks with the app.

Example 2 shows a review for each type of rogue story.

Example 2: Types of Rogue Stories

Rogue Act
"This is a really good app if u want to spy on your spouse I found out my boyfriend was cheating on me great app I recommend this app"

Rogue Potential
"...May work well to spy on the kids by 'accidentally' leaving iPhone in secret place."

We also found three types of reviewers writing rogue stories. First, reviewers who are *victims*: they state their concerns and grievances, including frustration at the loss of privacy. Second, reviewers who are *abusers*: they admit to the rogue behavior and sometimes express their delight in it. Third, reviewers are *third persons*: they report on others misusing the app or the potential to misuse. Example 3 shows a review for each type of reviewer.

Example 3: Categories Based on Reviewer

Victim
"I hate this app so much! My mother is always questioning me and if I delete it she will ground me ...No one want their parents to stalk them!!"

Abuser
"I can spy on my child whenever i want its amazing he cant go anywhere without me knowing look."

Third Person
"...I don't feel like parents should track their kids AT ALL. everyone needs a little something called trust and if you don't have it then your kids will act out and have to become sneaky. This app is designed to track families and see everything just like the parent is with you at all times. I do have this app but only with my fiends and we don't stalk each-other we just use it to see where everyone's at. And Bc we are so close and we all wanted it we all got it."

Table III shows the count of stories for each type of reviewer. The third person writes most of the potential reviews (44 out of 47) because such cases are only possibilities and not acts, meaning the reviewer is neither a victim nor an abuser. Whereas, abusers write other three potential cases. Such reviews (by abusers) indicate possible threats with the reviewed app, but suggest other apps for better rogue functionalities. For rogue act reviews, we found abusers (219) and victims (120) writing a majority of stories, followed by third person (16).

TABLE III
COUNT OF STORIES FOR EACH REVIEWER TYPE.

Reviewer	Rogue Act	Rogue Potential
Victim	120	0
Abuser	219	3
Third Person	16	44

To sum up, reviews describe apps' rogue behavior and show how victims such as children, parents, and friends are abused.

III. THE IROGUE APPROACH

iROGUE consists of three phases. First, iROGUE predicts the alarmingness score of each app review (Section III-A). Second, iROGUE generates a rogue score for each app based on the alarmingness of its reviews. We selected a threshold on rogue score, above which apps are predicted as rogue (Section III-B). Third, iROGUE finds additional rogue apps by examining apps in other datasets of scraped reviews (Section III-C). Figure 1 shows an overview of the iROGUE approach. We envision iROGUE to be incrementally updated by adding alarming reviews, of newly found rogue apps. Moreover, apps that don't have reviews yet, will be identified rogue, when their new reviews arrive.

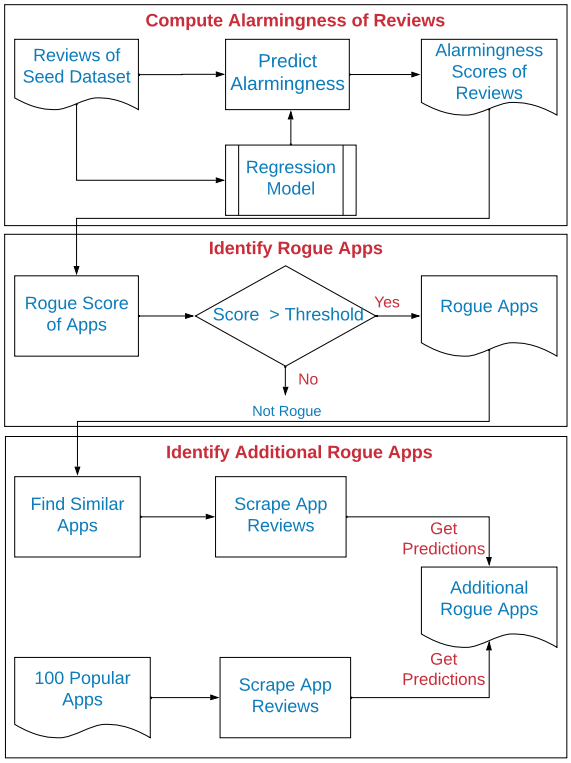


Fig. 1. Overview of iROGUE approach.

A. Computing Alarmingness of Reviews

Section II shows that app reviews reveal evidence of rogue behavior. However, identifying evidence in reviews is nontrivial, especially when an app receives a large number of reviews. Instead of binary classification of reviews, we introduce the alarmingness score that not only identifies relevant reviews but also ranks them based on the rogue behavior. To assess alarmingness of a review, we consider two factors: (i) the review's *convincingness* about rogue behavior and (ii) the *severity* of the reported rogue behavior. The alarmingness score of a review is the geometric mean of its convincingness and severity scores.

Reviews can vary in their claims about the rogue behavior. Some reviews report detailed rogue behavior, whereas some others are merely suspicion. The convincingness score measures how convincing the app review is in describing the rogue behavior. In Example 4, the first review is unrelated to rogue behavior and hence is not convincing. The second review describes the reviewer's suspicion on the app, which may or may not be true (slightly convincing). The third review (by an abuser) confirms the rogue behavior but lacks details of the rogue functionalities or victims. On the contrary, other reviews (in Example 4) are extremely convincing because they confirm rogue behavior along with mentioning the location feature, or how to set up devices, or the victims being stalked. Extremely convincing reviews include cases when the app is used for positive purposes (tracking family members or pets for safety) but has the potential to be misused in future. The reviews that are slightly, moderately, or extremely convincing are relevant to identifying rogue behavior. Assigning a convincingness score helps in ranking these reviews according to the strength of their claims.

The severity score measures the effect of rogue behavior on the victim. Example 5 shows range of reviews varying in severity. The first review is unrelated to rogue behavior. Thus, it is not severe. The second review shows that the rogue act is performed with consent, making this review a slightly severe case. The third review is written by the abuser and lacks the victim's perspective to analyze rogue effect. We assume such acts are performed without consent and consider them moderately severe. The fourth review describes the victim's misery. The victim even says "This app has truthfully ruined my teenage years" in the review, which gives solid evidence to be an extremely severe case. Moreover, in the fifth review, the victim complains that others can see when he was last active (also known as last seen information). This is the public information on each profile, but still the victim is uncomfortable with the access. App developers should be aware of such users' privacy expectations. However, such cases are still missed by the existing studies (Chatterjee et al., 2018; Roundy et al., 2020). Since app reviews discuss privacy expectations, we are able to identify such cases and rate them extremely severe.

Example 4: Varying Degree of Convincingness

1: Not Convincing
"It is such a great game, love it so much!"

2: Slightly Convincing
"Setup was a breeze. Quicktime 7 pro found it easily. Unfortunately, resolution seems much, much, lower than hoped. Video size can not be adjusted live. Hate to be a hater. May work well to spy on the kids by 'accidentally' leaving iPhone in secret place."

3: Moderately Convincing
"This app is perfect for stalking people. . . "

4: Extremely Convincing
"This app is awesome for our family to keep track of where everyone is at all times! (You can turn the location off too in case you want to be in stealth mode when buying Christmas presents too.) . . . Even our dog knows that the alert sound when a family member arrives home means . . . "

". . . I use it to spy on my dogs while I'm at work; so I use it for fun, nothing fancy. My iPad is my camera, and my iPhone is my viewer. . . . "

"bro this app is high key creepy. when i'm with my dad on his days my mom even mentions how she knew everything i was doing and it even made my dad creeped out. if you need this app then ngl yo wack. i don't want my mom stalking me."

Example 5: Varying Degree of Severity

1: Not Severe
"Love the graphics so far it is a great game"

2: Slightly Severe
"I love this app, just great because you can time your day accordingly, I like my girlfriend knowing where I am and I love stalking her, we have fun with it.. . ."

3: Moderately Severe
"This app is perfect for stalking people. . ."

4: Extremely Severe
"honestly if you want your kid to rebel against you even more, this is the app for you! This app has truthfully ruined my teenage years all because my mother now has a way of tracking me down 24/7. I couldn't do the normal teenage things because I was being stalked all day . . ."

". . . i want to share my last seen just to my family and my girlfriend not others. please add new feature in privacy that i can share my last seen to no body except my family and girl friend thanks soo much !"

We first rate the convincingness and severity of rogue behavior reported in the app reviews (Section III-A1). We then extract deep learning features from the reviews (Section III-A2). Leveraging the annotated set and extracted features, we evaluate various regression models and choose the best one (Section III-A3). Finally, we calculate the alarmingness score of an app review as the geometric mean of its predicted convincingness and severity scores.

1) Review Annotation: We selected 1,884 reviews from the seed dataset: 952 (set s_1) that contain at least one of our keywords and 932 reviews (set s_2) that do not contain any of those keywords. While preparing this annotation data, we exclude the reviewers' identifiers such as their usernames. Each selected review is rated for convincingness and severity on a four-point Likert scale (1: not, 2: slightly, 3: moderately, 4: extremely). For quality of annotations, we measure Inter Rater Reliability (IRR) via Intraclass Correlation Coefficient (ICC) Hallgren (2012). ICC is suitable for Likert scale ratings. Unlike other IRR measures such as Cohen's Cohen (1988) kappa, which are based on (all or nothing) agreement, ICC takes into account the magnitude of agreement (or disagreement) to compute IRR.

The annotation was conducted in three steps. First, two authors of this paper rated 599 of 1,884 selected reviews according to the initial set of annotation instructions. The initial instructions included definitions (of convincingness and severity scores) and examples corresponding to each point on the likert scale. In this step, for each annotator, we calculated the alarmingness scores of reviews using the convincingness and severity ratings. If the alarmingness scores computed for both the annotators were not at least three (median value on Likert scale), or both are not less than three, annotators resolved such cases via discussions. After discussing, the annotators produced the final set of annotation instructions. In the second step, the annotators followed the final instructions and rated 900 reviews. In this step of the annotation process, we achieved ICC of 0.9195 for convincingness and 0.9190 for severity. An ICC score in the range 0.75–1 indicates excellent agreement Hallgren (2012). In the third step, the remaining reviews were divided among the two annotators so that only one annotator rates each review.

For reviews that were rated by the two annotators, we computed the average convincingness and severity scores. This annotation study possessed minimal risk and was approved by the Institutional Review Board (IRB) of our university.

2) Extracting Deep Learning Features from App Review: We obtained the feature vector of each app review as follows:

Combine Sentences: Remove periods in each app review and combine all its sentences to form single sentence.

Text Preprocessing: Remove all punctuation marks, stop words Uysal and Gunal (2014), and our keywords, the latter because those keywords may correlate with reviews with higher scores and could create bias in the model.

Sentence Embedding: We leverage the Universal Sentence Encoder (USE) Cer et al. (2018) to extract embeddings for each app review. USE uses Deep Averaging Network (DAN) to provide a 512-dimension embedding for a long text. USE is trained on a large variety of natural language tasks with the aim of capturing the context. In our case, USE directly provides sentence level embeddings of an app review, by keeping the context intact. However, alternatives such as GLoVe Pennington et al. (2014) and Word2Vec Mikolov et al. (2013) lose such context. We leverage the pretrained USE network by using Tensorflow Hub Tensorflowhub.

3) Training Regression Model: We treat score prediction as a multi-target regression problem Borchani et al. (2015). Here, the 1,884 annotated reviews form the training set, and convincingness and severity are target variables, predicted using the extracted deep learning features.

We evaluated the performance of three regression models: support vector regression Basak et al. (2007), random forest Smith et al. (2013), and decision tree Xu et al. (2005), by ten-fold cross validation on our dataset. To mitigate bias of our keywords, we remove such keywords in the preprocessing step, so that the regression model learns from the the context of the review and not from specific keywords. Table IV shows average and standard deviation of mean squared error (MSE) Sammut and Webb (2010) in ten folds. The reported MSE is the combined MSE for two targets. The Support Vector Regressor (SVR) yields the smallest MSE, so we choose it for the subsequent phases of our approach.

TABLE IV
PERFORMANCE OF THREE REGRESSION MODELS ON TEN-FOLD CROSS VALIDATION.

Regression Model	Average MSE	Standard Deviation
Decision Tree	1.344	0.402
Random Forest	0.712	0.417
Support Vector	**0.625**	0.458

We use this trained model to predict convincingness and severity scores of all 11.57 million reviews in the seed dataset. The alarmingness of each review is calculated by taking the geometric mean of its predicted convincingness and severity. We use geometric mean because it ensures high alarmingness value only if both the convincingness and severity scores are high.

B. Identifying Rogue Apps

We produce an app's rogue score by aggregating the alarmingness scores of its reviews as follows.

Weighted Mean of Alarmingness: In general, for a rogue app, a small proportion of reviews report rogue behavior. Thus, we need to catch rogue apps using their few reviews that have high values on the alarmingness scale. Thus, we assign weights to reviews based on their alarmigness, as follows:

Defining score buckets: While annotating reviews, we defined levels of convincing reviews (not convincing to extremely convincing) and severe reviews (not severe to extremely severe) on a Likert scale. We also follow same levels on the alarmingness scale (1: not alarming to 4: extremely alarming). We define a score bucket between every consecutive level of alarmingness (not alarming to slightly, slightly alarming to moderately alarming, moderately alarming to extremely alarming). Table V shows how score buckets are formed using levels of alarmingness.

Assigning weights to score buckets: We have 11.57 million reviews in the seed dataset. Based on the alarmingness computation (Section III-A), we calculated the probability of a review falling in a score bucket. Since, the reviews reporting rogue behavior are less, probabilities in buckets 2 and 3 are less than that in bucket 1. We take inverse of these probabilities to get the weights for each score bucket. As a result, we assign higher weights to buckets 2 and 3 than to the bucket 1. Table V also shows the weight assigned to each score bucket.

TABLE V
SCORE BUCKETS FOR ALARMINGNESS.

Alarmingness Score Range	Alarmingness Level Range	Bucket	Bucket Weight
$[1, 2)$	Not alarming to Slightly	1	$2.29 \cdot 10^{-3}$
$[2, 3)$	Slightly to Moderately	2	$6.08 \cdot 10^{-2}$
$[3, 4]$	Moderately to Extremely	3	$9.36 \cdot 10^{-1}$

If a_1, a_2, …, a_n are the alarmingness scores of an app's reviews, and w_1, w_2, …, w_n are their respective weights (according to Table V), then, $W_{\text{alarmingness}}$, the weighted mean of alarmingness is given by:

$$W_{\text{alarmingness}} = \frac{a_1 * w_1 + a_2 * w_2 + \ldots a_n * w_n}{w_1 + w_2 + \ldots w_n}$$

The weighted mean of alarmingess ranges from 1 to 4.

Normalized Count: The weighted mean of alarmingness does not account for the count of reviews that report rogue behavior against an app. Suppose, *app A* has 15 reviews reporting rogue behavior and *app B* has 25 reviews reporting rogue behavior. If all reviews reporting rogue behavior have the same alarmingness score, the weighted means of the two apps would be the same. But, *app B* shows more evidence of rogue behavior and should have a higher rogue score than *app A*. Thus, we also consider the count of reviews. For each app, we calculate the number of reviews in bucket 3. We tried incorporating counts of other buckets, but it led to worse performance of the approach.

The minimum possible value of the count is zero. However, in some cases, counts can be high, leading to no definite upper limit. Thus, we normalize the counts of all the apps between one and four.

We want to assign high rogue score to apps that have high scores in both (i) weighted mean of alarmingness and (ii) normalized count. Thus, rogue score is computed as the geometric mean of these two values.

1) Selecting Threshold for Prediction: For each app in the seed dataset, iROGUE computes the rogue score. All apps are ranked in decreasing order of rogue scores. The apps with a score greater than a threshold are predicted rogue. To decide the correct threshold, we follow two steps: (i) label the ground truth of rogue apps and (ii) vary a threshold between certain values and choose the threshold which gives the best performance of iROGUE.

We create our ground truth by manually scrutinizing reviews. However, scrutinizing reviews of all 1,687 apps (seed dataset) is not feasible. Thus, first, we scrutinize the 50 most alarming reviews (with minimum score of two—at least slightly alarming) for apps with the highest 100 rogue scores. Second, we scrutinize reviews containing our keywords for these 100 apps. The first step is aligned with our approach since it checks top alarming reviews. However, the second step is neutral because it searches for evidence for the apps that iROGUE failed to identify through top alarming reviews. This way we mitigate the threat of bias, while curating ground truth for apps with the highest 100 rogue scores.

We label an app as rogue provided any of the scrutinized reviews report a rogue behavior. Table VI shows the types of reviews we consider indicative or otherwise of a rogue evidence. If the reviews of an app describe information access performed without the victim's knowledge or when the victim shows discomfort (first three reviews in Table VI), we consider the app as rogue. Also, some rogue apps are used for positive purposes such tracking family members for safety (fourth review in Table VI) but still possess a potential for future misuse. Similarly, apps used for tracking pets or other objects are considered rogue (fifth review). Reviews of some apps don't possess any misuse in present or in future, leading to their final label of not rogue (sixth review). Through manual inspection, we determine that of the 100 apps, 73 are rogue and 27 are not.

For the 100 apps, the rogue score varies between 1.74 and 3.60. We vary the threshold from 1.73 to 3.59 in steps of 0.01. Apps with a rogue score above the threshold are predicted rogue but not rogue otherwise. At each value of threshold, we report the recall, precision, and F1 score.

TABLE VI
INSTRUCTIONS FOLLOWED FOR LABELING ROGUE APPS.

Type of case	Subtype of case	Example	Evidence of rogue behavior
Tracking people's information	Without the victim's knowledge	*"Now that I can spy on my wife I will always know when she is cheating"*	Yes
Tracking people's information	With the victim's knowledge but with discomfort	*"Ok my mom got this for me and ... it's kinda creepy that this app was made so parent could basically stalk their kids."*	Yes
Tracking people's information	Public information but the victim is uncomfortable	*"I had someone cyberstalking and harassing me. Multiple attempts in every way shape and form were made to contact app-name to block and ban the stalker's account due to a concern for my well- being."*	Yes
Tracking people's information	Positive purpose	*"I love finding my family members. Wife was in bad car wreak and I was able to find her location using this app. Thank you!"*	Yes
Tracking pets or other objects		*"Wow! Day one and I'm stalking my puppet like a soccer mom that ran out of adderall! I'm very excited to use this to interact with my puppet while I'm at work and to check in on the dog walker!"*	Yes
Not related to information accessing		*"an absolutely amazing and very helpful app. i don't know how i would keep track of prayer times without it. love the app. thank u!!!"*	No

Table VII shows the performance achieved at specific thresholds. As we increase the threshold, the precision increases at the cost of recall. For rogue apps, a false negative costs more than a false positive because a false negative leaves a rogue app undetected, which can harm many victims, whereas a false positive causes only wasted effort in manual scrutiny. Hence, achieving high recall is more important than achieving high precision. Thus, from Table VII, we choose 1.73 threshold that gives the best recall of 100% at 73% precision. Since we fine-tune the threshold on the same seed dataset, we also check iROGUE's performance (using the chosen threshold) on the other dataset (Section III-C).

TABLE VII
CHOOSING AN APPROPRIATE THRESHOLD ACCORDING TO THE RECALL SCORES.

Threshold	Precision (%)	Recall (%)	F1 Score (%)
1.73	**73.00**	**100.00**	**84.39**
1.74	73.40	94.52	82.63
1.75	76.13	91.78	83.22
1.76	76.82	86.30	81.29
1.77	76.92	82.19	79.47
1.78	78.37	79.45	78.91
1.79	79.71	75.34	77.46
1.80	80.95	69.86	75.00
1.81	80.95	69.86	75.00
1.82	81.96	68.49	74.62
1.83	81.03	64.38	71.75
1.84	80.35	61.64	69.76
1.85	82.69	58.90	68.80
1.86	83.33	54.79	66.11
1.87	82.97	53.42	65.00

2) Performance of Baseline Methods: On the seed dataset, we also check the performance of baseline methods described below.

Our Keywords on App Description. We search for the presence of one of our keywords (*spy*, *stalk*, and *stealth*) in app descriptions. Apps whose descriptions contain any of these keywords are predicted rogue, whereas other apps are predicted as not rogue.

Example 6: Rogue App from Seed Dataset

App: Find My Family & Friends Life360
Rogue Score: 3.60/4.00

Alarming Review 1
Alarmingness: 4.00/4.00
Date of Review: 2019-11-28
"...Such a terrible thing for unaware parents to use. Most parents think teens don't need privacy and they constantly need to know where they are and what they're doing and who they're with at all times. This may make the parent feel at peace but what about the child? It's selfish of parents to not take into consideration of how the teen may feel about always having this app and the parent giving them a very stalkish feeling, it's very uncomfortable."

Extended Keywords on App Description. We identify additional relevant keywords by extracting verbs through Part-Of-Speech (POS) tagging Manning (2011). POS tagging marks every word in a sentence to an appropriate part of speech (verb, noun, adjective, and so on). Applying this process on descriptions of 73 rogue apps (from the ground truth) produced 145 verbs, out of which six (*track*, *monitor*, *locate*, *control*, *stolen*, *lost*) we selected as relevant to rogue behavior. The verbs: *stolen* and *lost* are relevant because they describe the apps that are used to find a misplaced phone, which indicates an ability to track another device. We

Example 7: Rogue App from Seed Dataset

App: OurPact Jr. Child App OurPact
Rogue Score: 2.47/4

Alarming Review 1
Alarmingness: 4.00/4.00
Date of Review: 2018-06-19
"...however this app shuts down almost everything and can see every text and website you've visited. now, i haven't done anything bad online (recently), but i find that a little creepy and honestly an invasion of privacy. no wonder this app has such crappy reviews. also, i used to have way more apps than i do now. because my parents now have the ability to restrict apps that may be "inappropriate". i already have to ask permission to download apps, so if they were inappropriate my parents wouldn't let me download them. there's too many apps like this and i think kids need a break from all this crap on their devices."

Alarming Review 2
Alarmingness: 4.00/4.00
Date of Review: 2018-08-16
"This is a useless app that no parent need to install I pray for every child who has this app installed on their electronics some parents don't understand the modern society but that's okay (but not really) I'm only given 2 hours and writing this review is using up time WHICH IS NOT FRIKEN OK!!! I hate this hate this app and I hope every child that has had their device attacked by this installment hates this app as much as me. This app should never be okay to use its inappropriate and everybody's children who have this app installed are making there children ANTI-SOCIAL AND VERY NOT COOL. I have many reasons why this app is SOOOOOOO scaring and dreadful so if your reading and thinking about installing this on ur child's device DONT INSTALL IT because that will ruin their future."

extend our keywords by adding these six verbs. Apps whose description contain these keywords are predicted rogue.

T% Keyword Reviews. For each app, we compute the percentage of reviews containing our keywords. We set a threshold, T, on this percentage, above which apps are predicted rogue. In our evaluation, T takes the values of 0.3, 0.2, and 0.1, respectively.

Table VIII summarizes the precision, recall, and F1 scores of all baselines and our approach. Our keywords on the description predict only one rogue app, leading to 100% precision (highest among all). However, our keywords miss 72 rogue apps, which leads to the worst recall of 1.36%. Among all the baselines, keyword search on reviews with 0.1% threshold achieves the highest recall of 65.07%, which is much lower than iROGUE's recall value. iROGUE's better performance may be due to fine tuning iROGUE's threshold on the same seed dataset. Thus, we also compare iROGUE's performance with these baselines on the other dataset (Section III-C).

Examples 6 and 7 show alarming reviews of Find My Family & Friends App Life360 and OurPact Jr. Child App OurPact, which iROGUE correctly identifies as rogue. Both of them are dual-use apps. Find My Family & Friends is a safety app, but alarming reviews report parents misusing the tracking functionality on children, to which children are uncomfortable. Moreover, the alarming reviews of the OurPact Jr. Child App report that parents can monitor children's texts and visited websites by installing the app on the child's device. In Section IV, we discuss these rogue functionalities in detail.

TABLE VIII
PERFORMANCE (IN %) OF BASELINE METHODS AND IROGUE ON THE SEED DATASET. BOLD VALUE FOR A METRIC INDICATES THE HIGHEST SCORE AMONG ALL APPROACHES.

Method	Recall	Precision	F1
Our keywords on app descriptions	01.36	**100.00**	02.68
Extended keywords on app descriptions	61.64	80.35	69.76
0.3% keyword reviews	46.03	96.66	62.36
0.2% keyword reviews	50.79	96.96	66.66
0.1% keyword reviews	65.07	95.34	77.35
iROGUE	**100.00**	73.00	**84.39**

C. Identifying Additional Rogue Apps

The scoring part of our approach is not dependent on the choice of candidate apps and could be applied on any dataset of apps. To identify additional rogue apps, we applied iROGUE's first two phases on two datasets: (i) dataset of similar apps and (ii) dataset of 100 popular apps in the utilities category.

1) Similar Apps: We retrieved 975 apps (similar to 100 predicted apps from the seed dataset), using the Apple App Store's recommendations ("You May Also Like" section). Our motivation in using Apple's recommendations is that these apps should offer functionalities similar to those in 100 predicted apps. Out of the 975 apps, reviews of 896 apps were present on the Apple App Store, over the period 2008-08-13 to 2022-08-24. We obtained 2,652,678 reviews. These 896 apps along with their reviews form our *snowball dataset*, as shown in Table IX.

TABLE IX
APPS AND REVIEWS IN THE SNOWBALL DATASET.

Similar Apps	Reviews
896	2,652,678

We apply iROGUE's first two phases (described in Sections III-A and III-B) on the snowball dataset. iROGUE predicts 138 rogue apps. Examples 8 and 9 show alarming reviews of two such apps from the snowball dataset, Smart Family Companion App Smartfamily and Bark - Parental Controls app Barkpc.

In the snowball dataset, to curate the ground truth, we follow the same labeling process as described in Section III-B1, for the apps with the highest 200 rogue scores. That's how we label 132 apps as rogue.

Table X shows the performance of all baseline methods and iROGUE on the snowball dataset. Our keywords when used on descriptions predicts only one app as rogue, leading to the lowest recall. This is because, on Apple App Store Appstore, dual-use apps are not advertised using keywords: *spy*, *stalk*, and *stealth*. The same approach achieves 100% precision but high recall is desirable in the context of rogue apps.

On app descriptions, extended keywords perform better (68.18% recall at 85.71% precision) than our keywords due to commonly used words (such as *track*, *locate*) in app descriptions. Our keywords are applied on reviews (rows 3–5 in Table X), and discover evidence of rogue behavior. However, among all approaches, iROGUE yields the best recall of 77.27%. As we discussed, high recall is desirable than high precision, we conclude that iROGUE outperforms all other methods.

TABLE X
PERFORMANCE (IN %) OF BASELINE METHODS AND IROGUE ON THE SNOWBALL DATASET. BOLD VALUE FOR A METRIC INDICATES THE HIGHEST SCORE AMONG ALL APPROACHES.

Method	Recall	Precision	F1
Our keywords on app descriptions	00.75	**100.00**	01.48
Extended keywords on app descriptions	68.18	85.71	**76.26**
0.3% keyword reviews	41.66	91.66	57.29
0.2% keyword reviews	44.69	92.18	60.20
0.1% keyword reviews	51.51	88.31	65.07
iROGUE	**77.27**	73.91	75.55

2) 100 Popular Utility Apps: Surveillance apps that can be misused for spying fall under the "Utilities" category, making utilities an important category to scrutinize. We consider 100 popular utility apps that are mentioned on Apple App Store page Utilitiesappstore. Out of 100 apps, nine are already scrutinized either in the seed or snowball dataset. For the rest 91 apps, we retrieved 392,928 reviews, over the duration of 2008-10-18 to 2022-08-04.

iROGUE predicts only one app as rogue, which after reviews' scrutiny by us, comes out to be non-rogue. We also scrutinize 10 apps with the highest rogue scores, by reading their top 50 alarming reviews and reviews containing our keywords. But, none of them are actually rogue. Since the Apple App Store Appstore contains a wide variety of utility apps, the selected 91 apps contain subcategories such as payment, calculator, and television remote. As a result, no video surveillance or parental control app, which have high potential to be misused, are part of 91 apps. Thus, popular utility apps don't form a good candidate set for rogue apps. This problem can arise for any generalized set of apps under any category. Thus, an iterative process of checking similar apps (through iROGUE), to the already identified rogue apps, can accelerate identifying more and more rogue apps.

IV. UNCOVERING ROGUE FUNCTIONALITIES

We now uncover rogue functionalities that are found via app reviews. Identifying such functionalities can help both users and developers. App users can understand the risk associated with the app and developers can rectify apps to reduce such risks.

Example 8: Rogue from Snowball Dataset

App: Smart Family Companion Smartfamily
Rogue Score: 2.35/4.00

Alarming Review 1
Alarmingness: 4.00/4.00
Date of Review: 2020-03-15
"How is this even ethical? To put out an app in which you can completely control what's going on on someone else's phone? It's a huge privacy concern. To be honest, apps like this shouldn't exist. It's one thing to put control on a YOUNG CHILD'S phone (which can be done in settings easily) put to put this on an older kids phone is going to destroy trust. No parent should be able to see what their child is doing on their phone 24/7. It's borderline abusive."

Alarming Review 2
Alarmingness: 4.00/4.00
Date of Review: 2019-09-03
"... This app tracks every last thing your child does on their phone. As you can imagine, no 16 year old wants to have their own private life constantly exposed to you. Just because you are their parent, and you live together, doesn't mean that they have to share everything with you. Location, data usage, browsing history, etc. frankly aren't your business. That's their own private information that you don't need to know. Coming from a family with control issues, there is no better way to destroy your relationship with your children. I doubt anyone would want to be around someone who is constantly monitoring and controlling them ..."

For this study, we considered apps in the seed dataset with the 40 highest rogue scores. For each app, we manually analyzed its description and its 10 most alarming reviews discovered by iROGUE. An app's description provides basic knowledge about the app's functionalities and alarming reviews report misuse of such functionalities. Through this exercise, we discovered the following types of rogue functionalities:

Monitoring phone activities. Some apps monitor a victim's phone activities, such as browsing history and text messages. Such apps are installed on the victim's device and activities can be monitored on another synced device.

Audio or video surveillance. Some apps enable audio or video surveillance without the victim's knowledge. These apps listen, view, or record a victim's voice or actions and some of them need not be installed on the victim's phone.

Tracking location. Some Global Positioning System (GPS) apps enable tracking a victim's phone, with (forced consent) or without their knowledge.

Profile stalking. Some apps are misused for stalking of user profile or user content (such as images), making the victim uncomfortable of the information access.

Example 9: Rogue App from Snowball Dataset

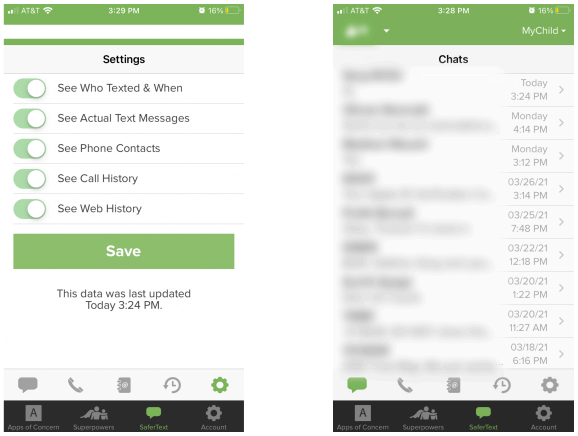
App: Bark - Parental Controls Barkpc
Rogue Score: 2.38/4.00

Alarming Review 1
Alarmingness: 4.00/4.00
Date of Review: 2020-05-28
"This app is unfair and invasion of privacy! Kids shouldn't be watched like this all kids cuss and this app tracks that and then snitches on you for it. I don't understand why someone can have so much doubt in their kids. Yes, I understand some kids do some really bad things that shouldn't be done, but if u raise your kids right and teach them right from wrong then you'd be able to trust them. One of my best friends has this app and she literally tells me how much she hates her parents. My friend has never even done anything and she has no reason for this app to be on her phone. I know the internet is dangerous but telling your kids it's dangerous honestly has a bigger effect. Maybe try other methods until it gets to the point of this app abolishing all their freedom and happiness."

Alarming Review 2
Alarmingness: 4.00/4.00
Date of Review: 2021-03-12
"If I could give this zero stars I would. This app is a total invasion of privacy and if you want to ruin your chances of having a relationship with your child, then get this. But if you are one of those parents who don't give your child/teen privacy you are not only hurting them but you are also hurting that bond and relationship with them. As a teen we don't want privacy because we are trying to hide something we just want privacy to be able to feel like our own person . . . "

Table XI shows these four types of rogue functionalities, and alarming reviews reporting them. Some reviews in Table XI are old (2014 or 2012), but we confirmed that similar concerns are being raised in the recent reviews of the same apps. For example, the Find My iPhone Findmyiphone app still lets its users see the location of the connected devices. Due to unequal power dynamics, the victim can be forced to connect to such apps and allow their device to be located.

We also verified the rogue behavior of the SaferKid Text Monitoring App by installing it on two devices: a parent's device (iOS version 14.4.1) and a child's device (Android version 11.0). Figure 2 shows the rogue features present in this app. Activities on the child's device can be monitored on the synced parent's device. Figure 2(a) shows SaferKid rogue functionalities such as monitoring text messages, web history, and call history. We verified each of these functionalities. Figure 2(b) shows the screen displaying all chats of the victim. Apps such as SaferKid are advertised as safety apps for children, but can be secretly or forcefully installed on another device to monitor user's activity. Not only parents, but any individual can misuse such apps by installing them on the victim's phone.



(a) The SaferKid Saferkid app provides multiple ways to monitor victim's phone activities.

(b) Using the SaferKid Saferkid app, victim's chats can be seen on the abuser's phone.

Fig. 2. Rogue functionalities in SaferKid Text Monitoring Saferkid.

V. Related Work

We describe previous works focusing on (i) spying through mobile apps, (ii) user privacy on social media, and (iii) NLP techniques to find apps' privacy issues.

A. Spying through Mobile Apps

Prior studies (Chatterjee et al., 2018; Roundy et al., 2020; Tseng et al., 2020; Freed et al., 2019, 2018; Zou et al., 2021; Tseng et al., 2021) investigate how technology is abused for spying. A major segment of this research deals only with IPS. Chatterjee et al. Chatterjee et al. (2018) identify IPS apps with carefully designed search queries and manual verification based on app information. They leverage information such as app descriptions and permissions. However, for dual-use apps, the actual usage deviates from the intended purpose shown in app descriptions. To identify such misuse, we focus on the evidence provided in app reviews. Moreover, the scope of rogue apps is broader than IPS apps.

Roundy et al. Roundy et al. (2020) focus on identifying apps used for phone number spoofing and message bombing, which lie outside the scope of rogue apps. Conversely, rogue apps include those that enable stalking public information, which are outside their scope. Roundy et al. use metadata such as installation data, to uncover spying apps that are installed on infected devices. However, we focus on evidence of rogue behavior present in app reviews, to uncover rogue apps. Roundy et al. rely upon Norton's security app NortonApp to determine which devices are infected. Thus, their approach would miss apps that a general user can leverage to spy.

Some prior studies focus on analyzing spyware apps or victims' experiences. Freed et al. Freed et al. (2019) present a qualitative analysis of victims' experiences, including their technology-related concerns. They report that security vulnerabilities were present in the phones of 14 out of 31 victims in their sample. Tseng et al. Tseng et al. (2020) study the IPS

TABLE XI
TYPES OF ROGUE FUNCTIONALITIES.

Rogue Functionality	App Example	Alarming Review
Monitoring phone activities	SaferKid Text Monitoring App Saferkid	*. . . Tracking things like social media, texts, and search history is just a complete disregard of privacy. You have to have trust in your kids . . . Apps like these shouldn't be allowed. IF YOU TRUST YOUR KID, DONT DOWNLOAD.* (Date: 2019-12-07)
Audio or video surveillance	Find My Kids: Parental control Findkids	*This app proves to have a invasion of privacy. Due to the fact if your kids was at a friends house and talking to his friends parents, this app records what is going on and is a invasion of privacy. If your child left their phone downstairs or anywhere and they are playing it can record private conversation between adult and is a unsafe . . .* (Date: 2019-01-16)
Tracking location	Find My iPhone Findmyiphone	*. . . It's supposed to be used to recover a lost phone, not to religiously stalk your children. . . . The fact that a mom actually installed this app onto her son's phone without his knowledge is flat out wrong. . . . If you're constantly monitoring your child 24/7, just imagine what your child will do when they go off to college. . . .* (Date: 2014-02-13)
Profile stalking	WhatsApp Messenger Whatsapp	*. . . However there is one negative about the App! The stalker look at the time stamp to monitor other people not nice please improve on that we need a sense of privacy from theses stalker*" (Date: 2012-11-21)

problem from the attackers' perspective. They analyze online forums in which attackers participate, propose a taxonomy of IPS tools and attacks. Tools may require physical device access, e.g., to install GPS trackers; or, they may rely on virtual access, e.g., through shared accounts of intimate partners. Attacks may include coercion or subterfuge, or may involve hiring another person to spy on someone. Havron et al. Havron et al. (2019) propose a consultation method, called clinical security, to help victims by discovering and removing spyware, and advising victims about security vulnerabilities in their phones. Moreover, Tseng et al. (2022) develop sociotechnical systems with feminist notions to help IPV survivors. Freed et al. Freed et al. (2018) survey spyware apps for intimate partners. They mention covert apps (also known as dual-use apps) that are capable of spying on victims but are not advertised as such. Fassl et al. (2022), through app reviews, study users' expectations from anti-stalkerware apps. They perform thematic analysis on 518 reviews of two apps and find a huge gap between users' perception and the actual abilities of such apps. All these studies along with others (Bellini et al., 2021; Zou et al., 2021; Tseng et al., 2021) are limited to IPS apps and not the broader set of rogue apps. Moreover, they do not consider cases when the victim is uncomfortable of the access (of public information) even if aware of it.

B. User Privacy

Prior works study risk of losing users' private information on online social media platforms and propose methods to mitigate such risk. Georgiou et al. Georgiou et al. (2017) protect users' privacy by giving warnings whenever a user may reveal sensitive attributes such as location or race present in social media posts. Mahmood and Desmedt et al. Mahmood and Desmedt (2012) claim that the Facebook friends of a user can access the user's private information in a cloaked manner. The study shows that it is possible to stalk and target victims on Facebook. Mahmood and Desmedt et al. provide strategies to avoid such attacks. Reichel et al. Reichel et al. (2020) study the privacy perspective of users in developing countries. They interview 52 social media users in South Africa to understand their privacy beliefs. Reichel et al. conclude that many participants are concerned about other users being able to see their online posts and messages, instead of the private data collected by the app platform itself. Many participants admitted that unknown people (on WhatsApp and Facebook) stalked or harassed them. To combat these challenges, Reichel et al. provide recommendations to fulfill users' security needs in resource-constrained situations. Some studies contribute to uncovering privacy risks associated with shared images (Henne et al., 2013; Bo et al., 2014; Perez et al., 2017), which can contain bystanders (persons who are not prime subject of image) and are shared widely without bystanders' consent. Hasan et al. Hasan et al. (2020) leverage visual features to detect bystanders in images present in the Google open image dataset Kuznetsova et al. (2020).

These studies are applicable to specific social media platforms and not to all rogue apps. Moreover, they do not provide a framework to identify rogue apps and their functionalities.

C. Using Natural Language Processing

We present prior studies that apply NLP techniques for security and privacy of mobile apps.

Some previous works leverage app reviews and privacy policies to identify user's security and privacy issues. Nguyen et al. Nguyen et al. (2019) train a classifier to predict if an app review pertains to security and privacy concerns. Using regression analysis, they show that security and privacy related reviews play an important factor in predicting privacy related app updates. Besmer et al. Besmer et al. (2020) leverage app reviews to understand how users' perception of privacy is reflected in their sentiments about the app. They train a machine learning classifier to determine whether a review is privacy related. Further, they analyze the sentiments of reviews predicted as privacy related. Harkous et al. Harkous et al. (2018) propose a privacy-centric language model to extract useful information from long privacy policies. The extracted information helps users understand how apps collect and manage users' personal information. To train the language model, they leverage 130,000 privacy policies. The trained model extracts both high-level and fine-grained details from policies. However, these studies focus on how an app which can steal a user's information. In contrast, we focus on the privacy of a victim (user or third party) with respect to another user.

Some prior works distinguish between the actual and the expected behavior (from user's perspective) of an app, by using textual sources such as descriptions and privacy policies. Gorla et al. Gorla et al. (2014) identify which apps deviate from their descriptions, by extracting topics from app descriptions, using Latent Dirichlet Allocation (LDA), and clustering apps based on those topics. For each cluster, Gorla et al. find outliers with respect to apps' APIs usage. Qu et al. Qu et al. (2014) and Pandita et al. Pandita et al. (2013) use NLP techniques on app descriptions and find disparities between app descriptions and functionalities. Zimmeck et al. Zimmeck et al. (2017) propose an automated system to find Android apps' compliance with their privacy policies. They combine static code analysis and machine learning to uncover inconsistencies between privacy policies and app source code. Out of 9,050 apps, Zimmeck et al. find that 17% of apps collect sensitive information such as location, but do not mention it in their privacy policies. All these works address expectation violation when the app developer has malicious intentions. However, in our work, we address expectation violation when an app user has malicious intentions to spy or stalk. To the best of our knowledge, iROGUE is the first automated system to identify rogue apps.

VI. DISCUSSION

We proposed iROGUE, an approach to automatically analyze app reviews for detection of rogue apps and rogue functionalities. iROGUE, first, predicts alarmingness of reviews, followed by rogue score for each app. In total, iROGUE predicts 239 rogue apps (100 and 139) from multiple sources, leading to the best recall, as compared to other baseline methods. We have also shared the identified rogue apps along with their reviews, to the Apple App Store. The platform will investigate these apps and will reaching out to the developers for correcting functionalities in their apps.

Below, we describe our data availability, threats to validity, and promising future directions.

A. Data Availability

Upon acceptance of this paper, we will release the complete list of all identified rogue apps. We will also release the scraped reviews as an open dataset. Our work is reproducible; we used Tensorflow Hub Tensorflowhub for extracting features and Scikit-learn library Pedregosa et al. (2011) for training the regression model, which will make available with the dataset.

B. Threats to Validity

We now discuss the threats we identify in our work. The identified threats are of two types: (i) the threats that we mitigate; and (ii) the threats that still remain.

1) Threats Mitigated: We mitigated the following threats to validity. First, reviews in the set s_1 contained our keywords. This may create a bias in the model to predict high scores for only reviews having our keywords. To mitigate this threat, we removed our keywords before training the model. This helped the model to learn from the context and not from specific keywords (described in Section III-A2). Second, review annotation by crowd workers could yield incorrectly rated reviews, because of their inability to understand the problem well. Thus, two authors of this paper annotate the whole training data. Third, the ground truth (of rogue apps) could be biased if it was formed only using top alarming reviews. We mitigated this bias by scrutinizing reviews containing our keywords, which can contain evidence missed by alarming reviews

2) Threats Remaining: Now, we describe the threats that still remain in our work. First, we investigate only a few thousand apps, which may not be representative of all apps on the Apple App Store Appstore. The performance of iROGUE may vary while testing it on all apps of the Apple App Store. Second, we target apps and their reviews only on Apple's App Store. Upon deployment on other app stores, the performance of our approach can differ. Third, if an app distribution platform does not have a similarity recommendation, iROGUE may have to be applied on all apps—a computationally expensive task. However, in such cases, iROGUE can be prioritized for the apps that are flagged by app users (victims). Fourth, some negative reviews (about rogue behavior) may be written by the app's competitors. Identifying such fake reviews is out of the scope of our study.

C. Limitations and Future Directions

We identify following limitations of this work. Each limitation gives rise to possible future work. First, iROGUE may miss some rogue apps if they do not have alarming reviews at the time of analysis, possibly because they are new apps. However, such rogue apps can be identified as soon as alarming reviews begin to arrive. By leveraging the current evidence, iROGUE helps protect future users and third parties. A possible extension for iROGUE would be to include other information sources such as privacy policies, to identify rogue apps ahead of time.

Second, uncovering rogue functionalities involves manual effort of inspecting top alarming reviews. A possible future direction is to automate this process.

REFERENCES

"AirBeam Video Surveillance App," https://apps.apple.com/us/app/airbeam-video-surveillance/id428767956.

"Apple App Store," https://www.apple.com/app-store/.

"Bark - Parental Controls," https://apps.apple.com/us/app/id1477619146.

D. Basak, S. Pal, and D. Patranabis, "Support vector regression," *Neural Information Processing – Letters and Reviews*, vol. 11, Nov. 2007.

R. Bellini, E. Tseng, N. McDonald, R. Greenstadt, D. McCoy, T. Ristenpart, and N. Dell, "So-called privacy breeds evil: Narrative justifications for intimate partner surveillance in online forums," *Proceedings of the ACM on Human-Computer Interaction*, vol. 4, no. CSCW3, Jan. 2021.

A. R. Besmer, J. Watson, and M. S. Banks, "Investigating user perceptions of mobile app privacy: An analysis of user-submitted app reviews," *International Journal of Information Security and Privacy (IJISP)*, vol. 14, no. 4, pp. 74–91, Oct. 2020.

C. Bo, G. Shen, J. Liu, X.-Y. Li, Y. Zhang, and F. Zhao, "Privacy.tag: Privacy concern expressed and respected," in *Proceedings of the 12th ACM Conference on Embedded Network Sensor Systems*. New York: Association for Computing Machinery, Nov. 2014, pp. 163–176.

H. Borchani, G. Varando, C. Bielza, and P. Larrañaga, "A survey on multi-output regression," *Wiley Interdisciplinary Reviews Data Mining and Knowledge Discovery*, vol. 5, no. 5, pp. 216–233, Sep. 2015.

D. Cer, Y. Yang, S. Kong, N. Hua, N. Limtiaco, R. S. John, N. Constant, M. Guajardo-Cespedes, S. Yuan, C. Tar, Y. Sung, B. Strope, and R. Kurzweil, "Universal sentence encoder," *CoRR*, vol. abs/1803.11175, pp. 1–7, 2018.

R. Chatterjee, P. Doerfler, H. Orgad, S. Havron, J. Palmer, D. Freed, K. Levy, N. Dell, D. McCoy, and T. Ristenpart, "The spyware used in intimate partner violence," in *Proceedings of the 39th IEEE Symposium on Security and Privacy (SP)*. San Francisco, CA, USA: IEEE Press, May 2018, pp. 441–458.

J. Cohen, *Statistical Power Analysis for the Behavioral Sciences*, 2nd ed. Hillsdale, New Jersey: Lawrence Erlbaum Associates, 1988.

M. Fassl, S. Anell, S. Houy, M. Lindorfer, and K. Krombholz, "Comparing user perceptions of Anti-Stalkerware apps with the technical reality," in *Proceedings of the Symposium on Usable Privacy and Security (SOUPS)*. Boston, MA: USENIX Association, Aug. 2022, pp. 135–154.

"Find My Kids: Parental control," https://apps.apple.com/us/app/id994098803.

"Find My iPhone App," https://apps.apple.com/us/app/find-my-iphone/id376101648.

D. Freed, J. Palmer, D. Minchala, K. Levy, T. Ristenpart, and N. Dell, ""A Stalker's Paradise": How intimate partner abusers exploit technology," in *Proceedings of the CHI Conference on Human Factors in Computing Systems*. New York: Association for Computing Machinery, Apr. 2018, pp. 1–13.

D. Freed, S. Havron, E. Tseng, A. Gallardo, R. Chatterjee, T. Ristenpart, and N. Dell, "Is my phone hacked? analyzing clinical computer security interventions with survivors of intimate partner violence," *Proceedings of the 17th ACM Conference on Human-Computer Interaction*, vol. 3, pp. 1–24, 2019.

T. Georgiou, A. E. Abbadi, and X. Yan, "Privacy cyborg: Towards protecting the privacy of social media users," in *Proceedings of 33rd International Conference on Data Engineering (ICDE)*. San Diego: IEEE Computer Society, Apr. 2017, pp. 1395–1396.

A. Gorla, I. Tavecchia, F. Gross, and A. Zeller, "Checking app behavior against app descriptions," in *Proceedings of the 36th International Conference on Software Engineering*. New York, NY, USA: Association for Computing Machinery, May 2014, pp. 1025–1035.

K. A. Hallgren, "Computing inter-rater reliability for observational data: An overview and tutorial," *Tutorials in Quantitative Methods for Psychology*, vol. 8, no. 1, pp. 23–34, 2012.

H. Harkous, K. Fawaz, R. Lebret, F. Schaub, K. G. Shin, and K. Aberer, "Polisis: Automated analysis and presentation of privacy policies using deep learning," in *Proceedings of 27th USENIX Security Symposium*. Baltimore: USENIX Association, Aug. 2018, pp. 531–548.

R. Hasan, D. Crandall, M. Fritz, and A. Kapadia, "Automatically detecting bystanders in photos to reduce privacy risks," in *IEEE Symposium on Security and Privacy (SP)*. Los Alamitos: IEEE Computer Society, May 2020.

S. Havron, D. Freed, R. Chatterjee, D. McCoy, N. Dell, and T. Ristenpart, "Clinical computer security for victims of intimate partner violence," in *Proceedings of the 28th USENIX Security Symposium*. Santa Clara: USENIX Association, Jan. 2019, pp. 105–122.

B. Henne, C. Szongott, and M. Smith, "Snapme if you can: Privacy threats of other peoples' geo-tagged media and what we can do about it," in *Proceedings of the 6th ACM Conference on Security and Privacy in Wireless and Mobile Networks*. New York: Association for Computing Machinery, May 2013, pp. 95–106.

A. Kuznetsova, H. Rom, N. Alldrin, J. Uijlings, I. Krasin, J. Pont-Tuset, S. Kamali, S. Popov, M. Malloci, A. Kolesnikov, T. Duerig, and V. Ferrari, "The open images dataset v4: Unified image classification, object detection, and visual relationship detection at scale," *International Journal of Computer Vision*, vol. 128, pp. 1–26, 2020.

"Life360 App," https://apps.apple.com/us/app/life360-find-family-friends/id384830320.

S. Mahmood and Y. Desmedt, "Your facebook deactivated friend or a cloaked spy," in *Proceedings of the 33rd IEEE International Conference on Pervasive Computing and Communications Workshops (PerCom Workshops)*. Los Alamitos: IEEE Computer Society, Mar. 2012, pp. 367–373.

C. D. Manning, "Part-of-speech tagging from 97% to 100%: Is it time for some linguistics?" in *Proceedings of the Computational Linguistics and Intelligent Text Processing*. Berlin, Heidelberg: Springer, Feb. 2011, pp. 171–189.

T. Mikolov, I. Sutskever, K. Chen, G. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," in *Proceedings of the 26th International Conference on Neural Information Processing Systems - Volume 2*, ser. NIPS. Lake Tahoe, Nevada: Neural Information Processing Systems Foundation, Dec. 2013, pp. 3111–3119.

G. A. Miller, "Wordnet: A lexical database for english," *Communications of the ACM*, vol. 38, no. 11, pp. 39–41, Nov. 1995.

D. C. Nguyen, E. Derr, M. Backes, and S. Bugiel, "Short text, large effect: Measuring the impact of user reviews on android app security & privacy," in *Proceedings of the 40th IEEE Symposium on Security and Privacy (SP)*. San Francisco: IEEE Computer Society, May 2019, pp. 555–569.

"Nortan Mobile Security App," https://buy-static.norton.com/norton/ps/bb/ushard/4up_mnav05w_us_en_fl_tw_branded_mix-n360.html.

"OurPact Jr. Child App," https://apps.apple.com/us/app/id1127917970.

R. Pandita, X. Xiao, W. Yang, W. Enck, and T. Xie, "WHY-

PER: Towards automating risk assessment of mobile applications," in *Proceedings of the 22nd USENIX Security Symposium*. Washington, D.C., USA: USENIX Association, Aug. 2013, pp. 527–542.

F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and Édouard Duchesnay, "Scikit-learn: Machine learning in python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

J. Pennington, R. Socher, and C. Manning, "GloVe: Global vectors for word representation," in *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Doha, Qatar: Association for Computational Linguistics, Oct. 2014, pp. 1532–1543.

A. J. Perez, S. Zeadally, and S. Griffith, "Bystanders' privacy," *IT Professional*, vol. 19, no. 3, pp. 61–65, 2017.

"Google Play Store," https://play.google.com/store/apps.

"PyDictionary," https://pypi.org/project/PyDictionary/.

"Thesaurus," https://pypi.org/project/py-thesaurus/.

Z. Qu, V. Rastogi, X. Zhang, Y. Chen, T. Zhu, and Z. Chen, "Autocog: Measuring the description-to-permission fidelity in android applications," in *Proceedings of the ACM SIGSAC Conference on Computer and Communications Security*. New York, NY, USA: Association for Computing Machinery, Nov. 2014, pp. 1354–1365.

J. Reichel, F. Peck, M. Inaba, B. Moges, B. S. Chawla, and M. Chetty, "'i have too much respect for my elders': Understanding south african mobile users' perceptions of privacy and current behaviors on facebook and whatsapp," in *Proceedings of the 29th USENIX Security Symposium*. Virtual: USENIX Association, Aug. 2020, pp. 1949–1966.

K. A. Roundy, P. B. Mendelberg, N. Dell, D. McCoy, D. Nissani, T. Ristenpart, and A. Tamersoy, "The many kinds of creepware used for interpersonal attacks," in *Proceedings of the 41th IEEE Symposium on Security and Privacy (SP)*. Los Alamitos: IEEE Computer Society, May 2020, pp. 753–770.

"SaferKid Text Monitoring App," https://apps.apple.com/us/app/saferkid-text-monitoring-app/id1143802529.

C. Sammut and G. I. Webb, Eds., *Mean Squared Error*. Boston, MA: Springer US, 2010, pp. 653–653.

"Smart Family Companion App," https://apps.apple.com/us/app/smart-family-companion/id1352914754.

P. F. Smith, S. Ganesh, and P. Liu, "A comparison of random forest regression and multiple linear regression for prediction in neuroscience," *Journal of Neuroscience Methods*, vol. 220, pp. 85–91, Oct. 2013.

"Tensorflow Hub," https://www.tensorflow.org/hub.

"3Fun: Threesome & Swingers App," https://apps.apple.com/app/id1164067996.

E. Tseng, R. Bellini, N. McDonald, M. Danos, R. Greenstadt, D. McCoy, N. Dell, and T. Ristenpart, "The tools and tactics used in intimate partner surveillance: An analysis of online infidelity forums," in *Proceedings of the 29th USENIX Security Symposium*. Virtual: USENIX Association, Aug. 2020, pp. 1893–1909.

E. Tseng, D. Freed, K. Engel, T. Ristenpart, and N. Dell, "A digital safety dilemma: Analysis of computer-mediated computer security interventions for intimate partner violence during covid-19," in *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*. Virtual: Association for Computing Machinery, 2021, pp. 1–17.

E. Tseng, M. Sabet, R. Bellini, H. K. Sodhi, T. Ristenpart, and N. Dell, "Care infrastructures for digital security in intimate partner violence," in *Proceedings of the 2022 CHI Conference on Human Factors in Computing Systems*. New Orleans: Association for Computing Machinery, 2022.

"Popular utilities apps," https://apps.apple.com/us/genre/ios-utilities/id6002.

A. K. Uysal and S. Gunal, "The impact of preprocessing on text classification," *Information Processing & Management*, vol. 50, no. 1, pp. 104–112, 2014.

"WhatsApp Messenger App," https://apps.apple.com/us/app/whatsapp-messenger/id310633997.

M. Xu, P. Watanachaturaporn, P. K. Varshney, and M. K. Arora, "Decision tree regression for soft classification of remote sensing data," *Remote Sensing of Environment*, vol. 97, no. 3, pp. 322–336, 2005.

S. Zimmeck, Z. Wang, L. Zou, R. Iyengar, B. Liu, F. Schaub, S. Wilson, N. Sadeh, S. Bellovin, and J. Reidenberg, "Automated analysis of privacy requirements for mobile apps," in *Proceedings of the Network and Distributed System Security Symposium*. San Diego: Korea Society of Internet Information, Feb. 2017.

Y. Zou, A. McDonald, J. Narakornpichit, N. Dell, T. Ristenpart, K. Roundy, F. Schaub, and A. Tamersoy, "The role of computer security customer support in helping survivors of intimate partner violence," in *Proceedings of the 30th USENIX Security Symposium*. Virtual: USENIX Association, 2021, pp. 429–446.