Erasing Labor with Labor: Dark Patterns and Lockstep Behaviors on Google Play

Ashwin Singh
IIIT Hyderabad, India
ashwin19.iiith@gmail.com

Arvindh Arun
IIIT Hyderabad, India
arvindh.a@research.iiit.ac.in

Pulak Malhotra
IIIT Hyderabad, India
pulak.malhotra@students.iiit.ac.in

Pooja Desur
IIIT Hyderabad, India
pooja.desur@students.iiit.ac.in

Ayushi Jain
IIIT Delhi, India
ayushi19031@iiitd.ac.in

Dueng Horng Chau
Georgia Institute of Technology, USA
polo@gatech.edu

Ponnurangam Kumaraguru
IIIT Hyderabad, India
pk.guru@iiit.ac.in

ABSTRACT
Google Play’s policy forbids the use of incentivized installs, ratings, and reviews to manipulate the placement of apps. However, there still exist apps that incentivize installs for other apps on the platform. To understand how install-incentivizing apps affect users, we examine their ecosystem through a socio-technical lens and perform a mixed-methods analysis of their reviews and permissions. Our dataset contains 319K reviews collected daily over five months from 60 such apps that cumulatively account for over 160.5M installs. We perform qualitative analysis of reviews to reveal various types of dark patterns that developers incorporate in install-incentivizing apps, highlighting their normative concerns at both user and platform levels. Permissions requested by these apps validate our discovery of dark patterns, with over 92% apps accessing sensitive user information. We find evidence of fraudulent reviews on install-incentivizing apps, following which we model them as an edge stream in a dynamic bipartite graph of apps and reviewers. Our proposed reconfiguration of a state-of-the-art micro-cluster anomaly detection algorithm yields promising preliminary results in detecting this fraud. We discover highly significant lockstep behaviors exhibited by reviews that aim to boost the overall rating of an install-incentivizing app. Upon evaluating the 50 most suspicious clusters of boosting reviews detected by the algorithm, we find (i) near-identical pairs of reviews across 94% (47 clusters), and (ii) over 35% (1,687 of 4,717 reviews) present in the same form near-identical pairs within their cluster. Finally, we conclude with a discussion on how fraud is intertwined with labor and poses a threat to the trust and transparency of Google Play.

KEYWORDS
Google Play, Dark Patterns, Fraud, Labor

1 INTRODUCTION
Google Play lists over 2.89 million apps on its platform [17]. In the last year alone, these apps collectively accounted for over 111 billion installs by users worldwide [15]. Given the magnitude of this scale, there is tremendous competition amongst developers to boost the visibility of their apps. As a result, developers spend considerable budgets on advertising, with expenditure reaching 96.4 billion USD on app installs in 2021 [16]. Owing to this competitiveness, certain developers resort to inflating the reviews, ratings, and installs of their apps. The legitimacy of these means is determined by Google Play’s policy, under which the use of incentivized installs is strictly forbidden [7]. Some apps violate this policy by offering users incentive in the form of gift cards, coupons, and other monetary rewards in return for installing other apps; we refer to these as install-incentivizing apps. Past work [6] found that apps promoted on install-incentivizing apps are twice as likely to appear in the top charts and at least six times more likely to witness an increase in their install counts. While their work focuses on measuring the impact of incentivized installs on Google Play, our work aims to develop an understanding of how it affects the users of install-incentivizing apps. To this end, we perform a mixed-methods analysis of the reviews and permissions of install-incentivizing apps.

Our ongoing work makes the following contributions:

(1) We provide a detailed overview of various dark patterns present in install-incentivizing apps and highlight several normative concerns that disrupt the welfare of users on Google Play.

(2) We examine different types of permissions requested by install-incentivizing apps to discover similarities with dark patterns, with 95% apps requesting permissions that access restricted data or perform restricted actions.

(3) We show promising preliminary results in algorithmic detection of fraud and lockstep behaviors in reviews that boost overall rating of install-incentivizing apps, detecting near-identical review pairs in 94% of the 50 most suspicious review clusters.

(4) We release our dataset comprising 319K reviews written by 301K reviewers over a period of five months and 1,825 most relevant reviews with corresponding qualitative codes across 60 install-incentivizing apps. [14]

2 DATASET
We created queries by prefixing “install apps” to phrases like “earn money”, “win prizes”, “win rewards”, etc., and searched them on Google Play to curate a list of potentially install-incentivizing apps. Then, we proceeded to install the apps from this list on our mobile
Figure 1: Distribution and CDF plot of install count for the 60 shortlisted install-incentivizing apps that collectively account for over 160.5M installs. Eighty-five percent of these apps have 100K or more installs, demonstrating their popularity.

Figure 2: Network of apps showing labels of five apps that share the most reviewers with other apps. App ‘us.current.android’ shares 6.4K reviewers with other install-incentivizing apps.

3 QUALITATIVE ANALYSIS

To understand the various ways in which install-incentivizing apps affect their users, we performed qualitative analysis of their reviews. Unless a user expands the list of reviews, Google Play displays only the top four “most relevant” reviews under its apps. Owing to their default visibility, we sampled these reviews for all 60 apps over a one-month period, obtaining 1,825 unique reviews. Then, we adopted an inductive open coding approach to thematically code [10] these reviews. In the first iteration, all researchers independently worked on identifying high-level codes for these reviews which were then compared and discussed. During this process, we defined the ‘completion of offers on install-incentivizing apps’ as an act of labor by users and the ‘incentive promised for their labor’ as value. Then, we reached a consensus on four high-level themes: exploitation, UI challenges, satisfaction, and promotion, which we define below:

1. **Exploitation**: User invests labor but is unable to gain value.
2. **UI challenges**: User invests labor but the app’s UI makes it challenging for them to gain value.
3. **Satisfaction**: User invests labor and is able to gain value.
4. **Promotion**: User invests labor in promoting an app through their review, rating or a referral code to gain value.

While all themes were useful for capturing the inter-relationship between a user’s labor and its value, the first three themes were relatively more prevalent in our data. Next, we performed two iterations of line-by-line coding of reviews within the high-level themes where the researchers identified emerging patterns under each theme until the principle of saturation was established.

3.1 How Install-Incentivizing Apps affect Users

In this section, we describe our findings from the qualitative analysis to shed light on how install-incentivizing apps affect their users. More specifically, we elaborate on the commonalities and differences of patterns within high-level codes that we discovered using line-by-line coding to depict how labor invested by users in these apps is not only exploited but also leads to negative consequences for them as well as the platform.

3.1.1 Dark Patterns.

Dark patterns can be defined as tricks embedded in apps that make users perform unintended actions [2]. We find comprehensive descriptions of dark patterns present within install-incentivizing apps in reviews coded as ‘exploitation’ and ‘UI challenges’. These patterns make it difficult for users to redeem value for their labor. First, our low-level codes uncover the different types of dark patterns present in reviews of install-incentivizing apps. Then, we ground these types in prior literature [9] by utilizing lenses of both individual and collective welfare to highlight their normative concerns. The individual lens focuses on dark patterns that allow developers to benefit at the expense of users whereas the collective lens looks at users as a collective entity while examining expenses. In our case, the former comprises three normative concerns. First, patterns that enable developers to extract labor from users without compensating cause financial loss (I1) to users. Second, cases where the data of users is shared with third parties without prior consent, leading to invasion of privacy (I2). Third, when the information architecture
of apps manipulates users into making certain choices due to the induced cognitive burden (I3). The lens of collective welfare facilitates understanding of the bigger picture of install-incentivizing apps on Google Play by listing three additional concerns. Due to high competition (C1), some developers incorporate dark patterns in apps that empower them to ‘extract wealth and build market power at the expense of users’ [4] on the platform. In conjunction with their concerns at the individual level, they also pose a serious threat to the price transparency (C2) and trust in the market (C3) of Google Play. In Table 1, we show these different types of dark patterns mapped to their individual and collective normative concerns using sample reviews from our data.

### Table 1: Different types of dark patterns mapped to their individual (Financial Loss (I1), Invasion of Privacy (I2), Cognitive Burden (I3)) and collective (Competition (C1), Price Transparency (C2), Trust in the Market (C3)) normative concerns.

<table>
<thead>
<tr>
<th>High-Level Code</th>
<th>Low-Level Code</th>
<th>Review</th>
<th>Normative Concerns</th>
</tr>
</thead>
<tbody>
<tr>
<td>Exploitation</td>
<td>Withdrawal Limit</td>
<td>100000 is equal to 10 dollars. Just a big waste of time. You can not reach the minimum cashout limit.</td>
<td>✓</td>
</tr>
<tr>
<td></td>
<td>Cannot Redeem</td>
<td>Absolute scam. Commit time and even made in app purchases to complete tasks ... I have over 89k points that it refuses to cash out!</td>
<td>✓</td>
</tr>
<tr>
<td></td>
<td>Only Initial Payouts</td>
<td>Good for the first one week then it will take forever to earn just a dollar. So now I quit this app ...</td>
<td>✓</td>
</tr>
<tr>
<td></td>
<td>Paid Offers</td>
<td>In the task I had to deposit 50 INR in an app and I would receive 150 INR as a reward in 24 hrs. 5 days have passed and I get no reply to mail.</td>
<td>✓</td>
</tr>
<tr>
<td></td>
<td>Hidden Costs</td>
<td>Most surveys say that the user isn’t eligible for them, after you complete them! Keep in mind you may not be eligible for 90% of the surveys.</td>
<td>✓</td>
</tr>
<tr>
<td></td>
<td>Privacy Violations</td>
<td>Enter your phone number into this app and you’ll be FLOODED with spam texts and scams. I might have to change my phone number because I unwittingly ...</td>
<td>✓</td>
</tr>
<tr>
<td>UI Challenges</td>
<td>Too Many Ads</td>
<td>Pathetic with the damn ads! Nothing but ads!!! Money is coming but only pocket change. It’ll be 2022 before I reach $50 to cashout, if then.</td>
<td>✓</td>
</tr>
<tr>
<td></td>
<td>Progress Manipulation</td>
<td>I redownload the app since the app would crash all the time ... I logged in and guess what?? ALL MY POINTS ARE GONE... 12k points all gone...</td>
<td>✓</td>
</tr>
<tr>
<td></td>
<td>Permission Override</td>
<td>When you give it permission to go over other apps it actually blocks everything else on your phone from working correctly including Google to leave this review.</td>
<td>✓</td>
</tr>
</tbody>
</table>

4 QUANTITATIVE ANALYSIS

In this section, we ascertain findings from our qualitative analysis as well as reveal more characteristics about the behavior of install-incentivizing apps and their reviews. For the same, we examine the permissions requested by these apps to establish their relevance to the dark patterns discussed in Section 3.1.1, and perform anomaly detection on their reviews to build upon the evidence of fraud from Section 3.1.2.

4.1 Permissions in Install-Incentivizing Apps

App permissions support user privacy by protecting access to restricted data and restricted actions on a user’s device [5]. Most permissions fall into two protection levels as determined by Android, namely normal and dangerous, based on the risk posed to user privacy. Similarly, another distinction can be made between permissions that access user information and permissions that only control device hardware [3]. We leverage these categories in our
analysis to identify types of permissions prominent across install-
incentivizing apps. Figure 3 shows an UpSet plot [8] of different 
types of permissions present in install-incentivizing apps. First,
we observe that over 92% of apps comprise dangerous permissions 
that access user information. The most popular permissions in this 
category include 'modify or delete the contents of your USB stor-
age' (41 apps), 'read phone status and identity' (24 apps), 'access 
precise location' (19 apps) and 'take pictures and videos' (14 apps).
Second, despite being requested by relatively fewer apps, some 
permissions in this category enable an alarming degree of control 
over user information; for e.g. 'create accounts and set passwords'
(5 apps), 'add or modify calendar events and send email to guests 
without owners’ knowledge’ (3 apps) and 'read your contacts' (2 
apps). Third, 34% of install-incentivizing apps contain permissions 
that access dangerous hardware-level information, the most promi-

4.2 Lockstep Behaviors

In Section 3.1.2, we found evidence of install-incentivizing apps 
indulging in review and rating fraud. Thus, we build upon the same 
to investigate reviews of these apps for anomalous behaviors such 
as lockstep that are indicative of fraud. Specifically, we focus on 
detecting groups of reviews that exhibit similar temporal and rating 
patterns; for e.g. bursts of reviews on an app within a short period 
of time to boost its overall rating.

4.2.1 Modelling and Experimental Setup.

Given that reviews are a temporal phenomenon, we model them 
as an edge-stream \( E = \{e_1, e_2, \ldots\} \) of a dynamic graph \( G \). Each 
edge \( e_i \in E \) represents a tuple \((r_i, a_i, t_i)\) where \( r_i \) is a reviewer 
who reviews an app \( a_i \) at time \( t_i \) (see Fig 4). Groups of fraudulent 
reviewers may either aim to boost the overall rating of an install-
incentivizing app or sink the rating of a competitor app. Thus, we 
partition our edge stream into two sub-streams as follows:

\[
\begin{align*}
E_{\text{boost}} &= \left\{(r_i, a_i, t_i) \in E \mid \text{Score}(r_i, a_i) \geq R_{a_i}, |E_{\text{boost}}| = 215,759 \right\} \\
E_{\text{sink}} &= \left\{(r_i, a_i, t_i) \in E \mid \text{Score}(r_i, a_i) < R_{a_i}, |E_{\text{sink}}| = 103,439 \right\}
\end{align*}
\]

where \( \text{Score}(r_i, a_i) \in \{1, 2, 3, 4, 5\} \) is the score assigned by 
reviewer \( r_i \) to the app \( a_i \) and \( R_{a_i} \) denotes the overall rating of app \( a_i \). 
Next, we reconfigure a state-of-the-art microcluster anomaly detection 
algorithm Midas-F [1] for our use. In particular, we modify the 
definition of a microcluster to accommodate the bipartite nature of 
our dynamic graph. Given an edge \( e \in E \), a detection period \( T \geq 1 \) 
and a threshold \( \beta > 1 \), there exists a microcluster of reviews on an 
app \( a \) if it satisfies the following equation:

\[
\frac{c(e, (n + 1)T)}{c(e, nT)} > \beta \ 
\text{where} 
\frac{c(e, nT)}{c(e, nT)} = 
\left| \left\{ (r_i, a_i, t_i) \mid (r_i, a_i, t_i) \in E_{\text{boost}} \land (n - 1)T < t_i \leq nT \right\} \right|
\]

if \( e \in E_{\text{boost}} \) and vice versa for \( E_{\text{sink}} \). Depending on whether \( e \) 
is a boosting or sinking edge, \( c(e, nT) \) counts similar edges for the 
app \( a \) within consecutive detection periods \((n - 1)T \) and \( nT \). Values 
recommended by the authors are used for the remaining parameters 
\( a \) and \( \beta \). It is worth noting that our modification preserves its 
properties of (i) theoretical guarantees on false positive probability, 
and (ii) constant-time and constant-memory processing of new 
edges [1].

4.2.2 Analysis and Preliminary Results.

Midas-F follows a streaming hypothesis testing approach that de-
termines whether the observed and expected mean number of edges 
for a node at a given timestep are significantly different. Based on 
a chi-squared goodness-of-fit test, the algorithm provides anomaly 
scores \( S(e) \) for each edge \( e \) in a streaming setting. Upon computing 
anomaly scores for both sub-streams \( E_{\text{boost}} \) and \( E_{\text{sink}}, \) we visual-
ize their CDF with an inset box plot in Fig 5. It can be observed 
that \( E_{\text{boost}} \) exhibits more anomalous behavior than \( E_{\text{sink}} \). To ascer-
tain statistical significance of the same, we make use of Welch’s 
t-test for the hypothesis \( H_1 \colon S_{\mu}(E_{\text{boost}}) > S_{\mu}(E_{\text{sink}}) \). We infer 
that reviews that aim to boost the rating of an install-incentivizing
We extract the 50 most suspicious clusters of reviews from both sub-
('good app', 'very good app'), ('good earning app', 'very good for
), from such pairs across 47 (94%) clusters from 
patterns. Lastly, from the fifty most suspicious clusters, we find 
reviews want to camouflage as normal users in terms of their rating 
reason for this dissonance can be that reviewers writing these 
adjectives considering they assign the app a low rating. A potential 
find that these are all extremely short reviews with at most three 
to four words that comprise mostly of adjectives; for e.g., $E_{\text{boost}}$: 
('good app', 'very good app'), ('good earning app', 'very good for 
earning app'), ('best app', 'very best app') and $E_{\text{sink}}$: ('bad', 'very 
bad'), ('super', 'super'), ('nice', 'very nice'). It is surprising to see 
that all but four identical pairs from $E_{\text{sink}}$ contain only positive 
adjectives considering they assign the app a low rating. A potential 
reason for this dissonance can be that reviewers writing these 
reviews want to camouflage as normal users in terms of their rating 
patterns. Lastly, from the fifty most suspicious clusters, we find 
such pairs across 47 (94%) clusters from $E_{\text{boost}}$ and 21 (42%) clusters 
from $E_{\text{sink}}$. This demonstrates that the efficacy of our approach 
towards detecting lockstep behaviors is not only limited to the 
temporal and rating dimensions, but also extends to the content 
present in reviews.

5 DISCUSSION AND FUTURE WORK
Our current work sheds light on how lax implementation of Google 
Play’s policy on fraudulent installs, ratings and reviews empowers 
developers of install-incentivizing apps to deplete the trust and 
transparency of the platform. Through use of permissions that 
access restricted data and perform restricted actions, developers 
icorporate dark patterns in these apps to deceive users and extort 
labor from them in the form of offers. The second form of labor that 
we study in our work is the writing of fraudulent reviews. We find 
evidence of their presence qualitatively and show promising results 
in detecting them algorithmically. Both types of fraud (incentivized 
installs and reviews) are only made possible by the labor of users 
who are vulnerable or crowd-workers who are underpaid [11]. This 
ensures developers to extract profits as they get away with violating 
Google Play’s policies without any consequences or accountability. 
However, a question that remains unanswered is, if reviews under 
these apps describe exploitative experiences of users, what is it 
that facilitates their continued exploitation? For now, we can only 
conjecture that fraudulent positive reviews on install-incentivizing 
apps suppress ranks of reviews containing exploitative experiences 
of users. Whether the same holds true or not is a question that 
remains to be explored in our future work.

REFERENCES

Figure 5: CDF plot of anomaly scores for the two edge streams $E_{\text{boost}}$ and $E_{\text{sink}}$. Reviews that boost the overall rating of an install incentivizing app exhibit significantly more anomalous behavior than reviews that aim to bring it down.

Figure 6: A microcluster anomaly detected by the algorithm where three reviewers are boosting the overall rating of two install-incentivizing apps ‘Cashyy’ and ‘Appflame’ on the same day.


