



“You Are Deceived in the Pocket”: An Exploratory Study of Intrusive Advertisements in Mobile Applications

MIAOYING CAI, Nankai University, China

DONGSUN KIM*, Korea University, South Korea

LINGLING FAN*, Nankai University, China

XIANGYU ZHANG, Nankai University and Zhongguancun Academy, China

SEN CHEN, Nankai University, China

Mobile advertising has become the primary monetization module for the Android ecosystem. However, this growth is accompanied by increasingly complex *intrusive advertisements* that undermine user agency through sophisticated behavioral interference. Current research on intrusive advertising primarily targets web or in-browser environments, failing to address the unique in-app characteristics of mobile intrusive ads. Existing ad analysis tools struggle to distinguish voluntary human actions from forced interactions due to the absence of intent-aware modeling. Furthermore, existing marketplace policies and legal frameworks lack unified terminology and enforceable rules, leading to inconsistent oversight. In this paper, we conduct an exploratory study to systematically investigate and model these intrusive advertising behaviors.

We propose a formal taxonomy grounded in regulatory policies, platform governance guidelines, industry standards, and documented real-world intrusive advertising practices from major SDK providers. Based on it, we apply an automated pipeline to detect intrusive patterns. Our analysis of more than 6,000 apps confirms the prevalence of mobile intrusive ads. Additionally, we perform a comparative analysis of mainstream regulations, uncovering significant misalignments in policy standards across different jurisdictions. Our study establishes a critical, theoretical, and practical foundation for ecosystem governance, enabling more effective detection and evidence-based policy refinement.

CCS Concepts: • **Software and its engineering**; • **Human-centered computing**;

Additional Key Words and Phrases: Mobile advertisements, Intrusive ads, Empirical study, Mobile applications

ACM Reference Format:

Miaoying Cai, Dongsun Kim, Lingling Fan, Xiangyu Zhang, and Sen Chen. 2026. “You Are Deceived in the Pocket”: An Exploratory Study of Intrusive Advertisements in Mobile Applications. *Proc. ACM Softw. Eng.* 3, ISSTA, Article ISSTA095 (October 2026), 22 pages. <https://doi.org/10.1145/3832186>

1 Introduction

The mobile advertising market has experienced rapid growth in recent years, becoming a cornerstone of the modern digital economy [31]. Driven by the widespread adoption of mobile devices and the proliferation of mobile applications, advertising has emerged as a dominant monetization

*Corresponding authors.

Authors’ Contact Information: [Miaoying Cai](mailto:miaoyingcai@mail.nankai.edu.cn), Nankai University, DISec, NDST, College of Computer Science, Tianjin, China, miaoyingcai@mail.nankai.edu.cn; [Dongsun Kim](mailto:dongsun.kim@korea.ac.kr), Korea University, Department of Computer Science and Engineering, Seoul, South Korea, darksrw@korea.ac.kr; [Lingling Fan](mailto:linglingfan@nankai.edu.cn), Nankai University, DISec, NDST, College of Cryptology and Cyber Science, Tianjin, China, linglingfan@nankai.edu.cn; [Xiangyu Zhang](mailto:xiangyu.zhang@nankai.edu.cn), Nankai University, DISec, NDST, College of Computer Science, Tianjin and Zhongguancun Academy, Beijing, China, 1120240353@mail.nankai.edu.cn; [Sen Chen](mailto:sen.chen@nankai.edu.cn), Nankai University, DISec, NDST, College of Cryptology and Cyber Science, Tianjin, China, senchen@nankai.edu.cn.



This work is licensed under a Creative Commons Attribution 4.0 International License.

© 2026 Copyright held by the owner/author(s).

ACM 2994-970X/2026/10-ARTISSTA095

<https://doi.org/10.1145/3832186>

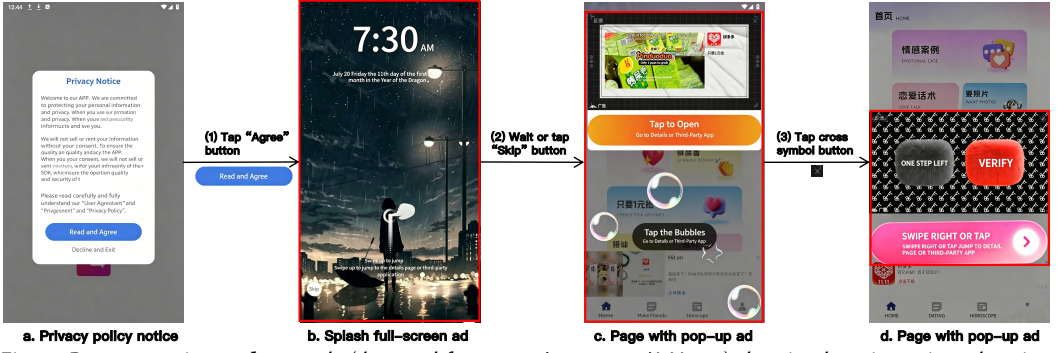


Fig. 1. Representative **real** example (detected from `com.hongyan.dijiuge`) showing how intrusive advertisements delay user access to intended functionality by interrupting essential interaction flows. After accepting the privacy agreement (state *a* and action (1)), a full-screen splash ad forces a wait or skip action (state *b*, action (2)). Once the home interface loads, functionality is again obstructed by a pop-up advertisement (state *c*) that must be manually dismissed to proceed. However, when the user clicks on the dismissal button (action (3)), another pop-up ad appears as state *d*.

strategy, enabling developers to offer services that are free or low-cost to end users. In particular, the Android ecosystem, which holds a dominant share of the global mobile operating system market [1], has become a major platform for mobile advertising at scale.

While this booming market drives the ecosystem’s growth, the interests of its various stakeholders are not always aligned. The mobile advertising ecosystem comprises multiple stakeholders: (1) advertisers who pay for user attention, (2) ad platforms and networks that facilitate delivery, (3) SDK providers who supply integration tools, (4) app developers who implement ads, and (5) end users who consume apps [38]. While advertisers, platforms, SDK providers, and developers all derive financial benefits from ad delivery, end users often bear the cost in the form of degraded usability, privacy risks, and distraction. This imbalance becomes particularly pronounced when advertising practices turn aggressive or deceptive, creating a fundamental tension between monetization objectives and users’ intended app interactions. Such practices are commonly referred to as **intrusive advertisements (ads)**, which broadly manifest as either in-app intrusive ads embedded within application flows or in-browser intrusive ads delivered through web-rendered interfaces.

Mobile users often suffer from intrusive ads as other stakeholders want to maximize their gains. As a result, users may be repeatedly interrupted, delayed, or forced into unintended interactions during normal app usage. For example, during an application launch, a user may face a multi-stage advertising sequence, as illustrated in Figure 1, which is a real case found in `com.hongyan.dijiuge`. After accepting the privacy agreement (state *a* and action (1)), a full-screen splash ad forces a wait or skip action (state *b*, action (2)). Once the home interface loads, functionality is again obstructed by a pop-up ad (state *c*) that must be manually dismissed to proceed. However, when the user clicks on the dismissal button (action (3)), another pop-up ad appears as state *d*. This example demonstrates how intrusive ads are strategically placed at critical entry and transition points, delaying user goals and embedding advertising interactions into essential navigation flows.

Existing research on intrusive ads has predominantly focused on the web context, leaving a significant gap in the understanding of the native mobile environment. To the best of our knowledge, only two studies have investigated intrusive ads [35, 37]. However, their scope is limited to in-browser ads, which operate under a fundamentally different mechanism than in-app advertising in mobile apps. In-browser ads are rendered as web elements within a browser engine, allowing detection through DOM analysis. In contrast, in-app ads delivered via embedded SDKs are tightly

coupled with the app’s native UI and lifecycle. They interact not only through touch inputs but also through system-level navigation and application state transitions. Therefore, although these works discuss the mobile domain, their web-centric methodologies cannot capture the complex interaction between ad displays and app functionality.

Despite extensive research on mobile advertising, the implications of in-app ads on user interaction remain insufficiently characterized. Existing work has largely centered on the interests of advertisers and platforms, emphasizing the detection of legitimate ad traffic. Complementary studies have examined adversarial and security-related concerns, including ad fraud [12, 23], malicious advertising [24, 25], and privacy violations [28, 30, 32]. While these efforts provide valuable economic and security insights, they overlook a gray area: ads that are technically “safe” and “valid” by industry standards, yet functionally intrusive to the user. This limitation arises from a focus on UI patterns rather than the integrity of user intent, resulting in methods that offer limited insight into how intrusive ads systematically undermine user agency and subvert user intent.

As a result, current regulatory and detection efforts face fundamental limitations due to a fragmented landscape. Marketplace policies and industry standards rely on subjective terminology without operational definitions [14–16], forcing manual review processes that do not scale and often miss dynamically triggered behaviors. Existing studies address isolated aspects of intrusive mobile ads, such as accessibility interference [18], there is a lack of comprehensive taxonomies and unified modeling frameworks to characterize ad triggers systematically. This leaves multiple stakeholders without clear guidance criteria to identify and mitigate intrusive behaviors.

To address these gaps, we conduct the first systematic investigation of intrusive advertising behaviors on the Android platform. We have pinpointed three major challenges. First, intrusive ads in mobile apps lack a unified characterization among different policies and standards, as existing descriptions remain ambiguous. We address this by constructing a formal taxonomy that maps intrusiveness to observable UI and interaction patterns. Second, many intrusive behaviors are deeply hidden in interaction flows and cannot be easily triggered through naive exploration. Guided by the taxonomy, we conduct dynamic exploration with targeted action injection and sequence modeling to elicit such hidden multi-stage behaviors actively. Third, there is a disconnection between user-perceived and triggering mechanisms. We bridge this gap by performing offline multi-modal analysis to correlate UI transitions, user actions, and ad contexts to reconstruct causal trigger chains.

We leverage the above principles to detect intrusive ads at scale and further analyze their potentially risky impact. Our large-scale empirical study confirms the prevalence of such intrusions in major Android markets. The results show that functionality-interrupting ads constitute the most prevalent category of mobile intrusive ads. In addition, we observe that intrusive ads tend to appear in combined forms, where multiple intrusive types are developed together to amplify their disruptive effect on user interaction. Beyond usability disruption, we further identify that risky content frequently accompanies intrusive ads. Through a policy-relevant app analysis, we confirm that mobile intrusive advertising is not merely a usability issue, but rather a policy-level concern. Furthermore, through a comparative analysis of existing platform policies and regional regulations, we highlight critical gaps and provide actionable insights for ecosystem refinement.

We aim to answer the following research questions:

- **RQ1:** [Taxonomy] What patterns characterize mobile intrusive advertising behaviors?
- **RQ2:** [Detection] How prevalent are they across the Android app ecosystem?
- **RQ3:** [Impact] What are the potential risks associated with intrusive advertising?
- **RQ4:** [Policy Alignment] To what extent do intrusive ad implementations align with existing policy frameworks?

In summary, we make the following contributions:

- **Comprehensive taxonomy.** We develop a taxonomy of mobile intrusive ad behaviors grounded in industrial standards and augmented through empirical analysis. We introduce a formalized representation that captures such variations in a unified manner, enabling consistent interpretation and practical operationalization.
- **Large-scale empirical study.** We systematically analyze more than 6,000 Android apps of markets. Our analysis reveals that intrusive advertising is prevalent and poses broader impact beyond usability concerns.
- **Policy gap analysis.** We present a comparative study of main regulations and platform policies, identifying inconsistencies between legal requirements and actual ad behaviors.

2 Background and Motivation

2.1 The Mobile Advertising Ecosystem

This section introduces the stakeholders in the mobile advertising ecosystem, clarifies intrusive advertising behaviors, and reviews existing policies. Understanding these foundations is critical for contextualizing the gap between stated policies and actual practices that our work addresses.

Advertisers pay to promote products or services through mobile ads, aiming to maximize return on investment (ROI) via models such as cost-per-impression (CPM), cost-per-click (CPC), or cost-per-action (CPA). While advertisers seek genuine user engagement, metric-driven incentives (e.g., high click-through rates) may unintentionally encourage intrusive practices, especially when engagement quality and user experience are not considered.

Ad platforms act as intermediaries between advertisers and developers, providing Ad SDKs, managing real-time auctions, distributing revenue, and enforcing policies. While platforms maintain guidelines against intrusive ads to preserve ecosystem health, aggressive ad delivery can still be financially attractive in the short term.

Application developers rely on in-app advertising as a primary monetization strategy, particularly for free apps, by integrating third-party Ad SDKs. Revenue is generated through ad views or interactions, creating incentives to increase ad exposure. Although developers control when and where ads are triggered, they often have limited control over ad creatives and runtime behaviors, leading to potential conflicts between monetization goals and user experience.

End users are the recipients of mobile ads but have the least control over ad experiences. In free apps, users must tolerate ads to access functionality. Unlike other stakeholders whose incentives align around maximizing engagement, users generally prefer minimal disruption and clear exit options, placing them at odds with prevailing economic incentives.

2.2 Intrusive Advertising Behaviors

Intrusive advertising refers to any ad delivery method or design that significantly disrupts the user's workflow or manipulates user interactions beyond the scope of voluntary engagement. Intrusiveness manifests through three core dimensions: (1) *UI Manipulation* exploiting visual design to deceive or obstruct users (e.g., obscured close buttons, fake functional buttons); (2) *Behavioral Disruption* interrupting intended functionality through temporal interference (e.g., forced viewing, hijacked buttons); (3) *Frequency and Persistence* creating constant interruption through excessive ad density relative to user actions or time.

Notably, intrusiveness does not necessarily equate to fraudulent or malicious content. For example, a legitimate ad provided by a major platform appearing every two minutes with a barely visible close button generates valid revenue for all parties, yet severely frustrates users. Conversely, click injection schemes operating silently in the background do not disrupt foreground activities. We

assume all analyzed user behaviors are genuine attempts by real people to accomplish legitimate tasks. When users struggle to close a “mini” button or accidentally trigger ads through normal device movements, these are legitimate experiences worthy of protection, not anomalies to filter out. This assumption fundamentally distinguishes our work from fraud detection, which must identify technical violations and distinguish real users from bots.

2.3 Industrial and Regulatory Standards

To maintain a sustainable ecosystem and user trust, various industry bodies and platform operators have established guidelines to regulate ad behaviors. Some policies, like *the Better Ads* [9] and *the Acceptable Ads Standards* [3], have been developed based on user research to identify the most displeasing ad experiences. For mobile apps, these standards discourage ads that unexpectedly interrupt content. *Google Play Store* [15] policies prohibit ads from interfering with device functionality, displaying in unexpected contexts, lacking clear dismissal options, and so on. Beyond these industry norms, regulatory frameworks across regions have imposed legally binding requirements to curb intrusive and non-transparent advertising. In this context, Ad SDK providers (e.g., Google AdMob [15], and Pangle SDK [10]) publish recommendations, including the discouraged ones, to align mobile ads implementations with policies and prevent developers from violating regulations.

2.4 Motivation

To understand the necessity of studying mobile intrusive ads, we first conducted a preliminary investigation across user communities, developer discussions, and regulatory policies. By connecting our real-world case (Figure 1) with broader observations, we identify the technical gaps in current tools and the unmet needs of different stakeholders.

2.4.1 Risk Characterization and Technical Gaps. Taking the cases in Figure 1 as a start point, we identify risks the users face and critical technical gaps that current tools fail to address.

- **Erosion of user agency and navigation integrity:** The most significant risk to the user is the loss of navigation integrity. In our case, action (3) illustrates how a user’s intent to “dismiss” is subverted to trigger another ad state. This creates an interaction trap where the user’s mental model of the app’s navigation is broken. Unlike typical security risks (e.g., data theft), intrusive ads work more subtly and damage users’ experience, further leading to a fundamental breakdown of trust in the mobile ecosystem.

- **Blindness of web-based detection tools:** Existing web-based intrusive ad detectors (e.g., AdHere [35]) rely on DOM structure analysis. However, mobile in-app intrusive ads are deeply embedded in the native view hierarchy and activity lifecycle. As seen in the case, the transition from state *b* to state *c* is not a simple element rendering but a triggered event within the app’s internal logic. Unlike in-browser ads, where elements can be located by CSS selectors, in-app intrusive ads are often dynamically injected by black-box SDKs or runtime mechanisms, rendering them invisible to DOM-based detection approaches.

- **Incomplete context in ad analysis:** Current ad analysis tools are primarily designed from the advertisers/ad platforms’ perspectives, focusing on protecting ad ecosystem integrity through traffic validation. While some efforts examine UI features, they lack a user-centric module to account for user intent. Consequently, they fail to distinguish voluntary actions from hijacked ones. In our case, when a “dismiss” click is subverted to trigger a new ad, these tools consider it is a successful engagement while ignoring its involuntary nature.

2.4.2 Community Needs. We examined user and developer communities like *StackOverflow* and *Reddit* to collect real-world complaints and implementation concerns related to mobile ads. We also reviewed documentation and compliance guides from popular advertising SDK providers, as

well as public reports and regulatory notices concerning intrusive ads. These materials collectively provide a multifaceted view of how intrusive ads affect different stakeholders across the mobile advertising ecosystem.

- **User perspective (The Victim).** For users, ads that disrupt the app's natural interaction flow are often perceived as annoying and intrusive. Users are particularly sensitive to ads that interrupt ongoing tasks, obscure essential content, or appear excessively during use. Such behaviors often lead to app uninstalls, reduced engagement, and overall distrust toward the application ecosystem.

- **Developer perspective (The Dilemma).** For developers, ads serve as a critical source of revenue. However, developers often have limited control over ad content and behavior, especially when integrating third-party Ad SDKs. Due to a lack of clear technique guidance, they are unable to verify the application's compliance. As a result, even well-intentioned implementations may inadvertently violate app store policies or user experience standards.

- **Ecosystem and regulatory perspective (The Fragmentation).** From a broader ecosystem standpoint, multiple stakeholders have introduced different standards and regulatory measures to define and restrict intrusive ads. The Coalition for Better Ads proposed the *Better Ads Standards* [9], which have been adopted by Google in its *Ad Experience Report* [16]. Major ad SDK providers have released their own compliance guidelines and publicized examples of improper integrations. Meanwhile, regulatory bodies such as China's Ministry of Industry and Information Technology (MIIT) [33] regularly publish violation cases of mobile apps with deceptive or intrusive behaviors. These initiatives, while valuable, remain fragmented and inconsistent in scope and terminology.

3 Taxonomy of Mobile Intrusive Ads

To systematically characterize the mobile intrusive advertising, we develop a comprehensive taxonomy grounded in three complementary sources: (1) platform and industry policies, (2) regional regulatory frameworks, and (3) documented violation cases reported in SDK compliance notices, regulatory enforcement disclosures, and user-reported complaints. Collectively, these sources span two complementary dimensions: *platform and industry standards* and *jurisdiction-specific regulations* from multiple regions (e.g., China and Korea), ensuring our taxonomy reflects both ecosystem-level governance norms and legally binding requirements.

The construction of this taxonomy follows a rigorous multi-stage process:

- **(1) Policy Analysis & Rule Extraction:** We systematically analyzed the collected documents to extract rules specifically governing *visual presentation* and *interaction behavior*.
- **(2) Case Distillation:** Recognizing that high-level policy language is often abstract and inefficient for operational detection, we therefore grounded each policy rule in concrete behavioral instances by collecting documented violation cases from SDK compliance guides, official enforcement notifications, and user community reports (e.g., Reddit, app store reviews). This inductive step allowed us to bridge the gap between normative policy intent and observable UI/interaction patterns, yielding a set of *atomic behavioral primitives* as the building blocks for classification.
- **(3) Synthesis & Classification:** The extracted behavioral primitives were iteratively grouped according to their underlying intrusive mechanisms. To evaluate the coverage of the taxonomy, we applied it to a pilot set of 20 real-world applications spanning diverse app categories and ad integration patterns.

All observed intrusive behaviors could be consistently mapped to one of the five categories, suggesting that the taxonomy achieves descriptive saturation with respect to currently documented policies and observed practices. Nevertheless, we acknowledge that emerging advertising techniques and evolving platform ecosystems may require future extensions.

While the taxonomy above characterizes intrusive behaviors at a conceptual level, policy descriptions and human-readable examples alone are insufficient for scalable automated auditing. To operationalize the taxonomy, we therefore formalize mobile app interaction as a state transition system $\mathcal{M} = (\mathcal{S}, \mathcal{E}, \delta)$. This formalization reframes intrusiveness not as a subjective annoyance, but as a *quantifiable violation of state transition logic and user intent*, enabling both systematic detection and evidence-based regulatory auditing.

3.1 Application State Model

We model an application execution as $\mathcal{M} = (\mathcal{S}, \mathcal{E}, \delta)$, where:

- $\mathcal{S}$ is the set of app states. We partition $\mathcal{S}$ into three distinct subsets:
 - S_{task} : **Functional States**, where the user engages in workflow (e.g., reading news).
 - S_{ad} : **Ad States**, where an ad is present and constitutes the current focus of interaction.
 - S_{target} : **Ad Target States**, representing the state reached as a target of an ad-trigger transition or navigation, including in-app landing pages or external contexts (e.g., browser, app store).
- $\mathcal{E}$ is the set of events triggering transitions. We distinguish user-initiated events:
 - e_{intent} : Explicit user interaction intended for app function (e.g., clicking “Next”).
 - e_{close} : Explicit user interaction intended to dismiss content (e.g., clicking “X”).
 - e_{gen} : generic navigation gestures without a specific target (e.g., scrolling, swiping).
 and environment-initiated events: e_{wait} (passage of time without user input) and e_{sensor} (sensor events, e.g., gyroscope tilt).
- $\delta : \mathcal{S} \times \mathcal{E} \rightarrow \mathcal{S}$ is the transition function mapping the state and an event to the next state.

Formal Definition.

Let $s_{ad} \in S_{ad}$ denote an advertisement state, and let $V(s_{ad})$ denote the set of visible UI elements in s_{ad} . For each element $v \in V(s_{ad})$:

- $\Phi(v)$: The visual semantics of element $v \in V$. This represents the function visually implied by the element (e.g., close, confirm, download). We denote the set of semantics of common app functions as Φ_{app} , and the semantics of exit as Φ_{exit} .
- $\delta(s_{ad}, interact(v))$: Actual transition triggered by interacting with v , where $interact(v) \in \mathcal{E}$.

3.2 Categories of Mobile Intrusive Ads

Based on existing documents and manual inspection of real-world apps, our categorization considers multiple dimensions of intrusiveness, including visual interface deception, exit obstruction, disruptive activation, spatial intrusion, and temporal intrusion.

3.2.1 [T1] Deceptive UI Ads. This category represents the form of visual manipulation, where the ad’s interface is designed to confuse or mislead. The four patterns within this category collectively undermine fundamental principles of honest interface design, specifically by subverting user expectations and misrepresenting UI semantics to induce involuntary engagement.

We denote v_{exit} as an element with the visual semantic meaning of exit. A state $s_{ad} \in S_{ad}$ is classified as a *Deceptive UI Ad* if the interface fails to provide a clear and perceivable exit or control mapping, manifested by at least one of the following conditions:

(T1.1) Exit Absence

- *Description.* The ad state is displayed (S_{ad}), but the set of interactive elements contains no element that triggers a return to the task state (S_{task}).

$$\nexists v_{exit} \in V(s_{ad}) : \delta(s_{ad}, interact(v_{exit})) \in S_{task}$$

- *Example.* A full-screen ad that lacks an “X” or “Skip” button, leaving users unable to dismiss the ad and interrupting normal app usage.

(T1.2) Exit Obfuscation

- *Description.* The ad state $s_{ad} \in S_{ad}$ contains an exit control v_{exit} , but its salience (perceptibility) is below the threshold (τ) due to poor visibility (e.g., low contrast, transparency and overlapping).

$$\exists v_{exit} \in V(s_{ad}) : \delta(s_{ad}, \text{interact}(v_{exit})) \in S_{task} \wedge \text{salience}(v_{exit}) < \tau_{sal}$$

- *Example.* A semi-transparent close button is placed in a corner, making it difficult for users to notice or accurately tap.

(T1.3) Exit Ambiguity

- *Description.* The ad state S_{ad} contains multiple UI elements that visually convey the semantic intent of exit, making it ambiguous which element safely returns the user to the original context.

$$\exists v_1, v_2 \in V(s_{ad}), v_1 \neq v_2 : \Phi(v_1) \in \Phi_{exit} \wedge \Phi(v_2) \in \Phi_{exit} \wedge$$

$$\delta(s_{ad}, \text{interact}(v_1)) \in S_{task} \wedge \delta(s_{ad}, \text{interact}(v_2)) \in S_{target}$$

- *Example.* Two visually similar close buttons are shown, where one correctly closes the ad, but the other opens the app store.

(T1.4) Misleading Affordance

- *Description.* Within the ad interface (s_{ad}), a deceptive UI element mimics the appearance of a common function (Φ_{app}), but interaction leads to redirect the user to an ad target state (S_{target}).

$$\exists v \in V(s_{ad}) : (\Phi(v) \in \Phi_{app}) \wedge (\delta(s_{ad}, \text{interact}(v)) \in S_{target})$$

- *Example.* A button labeled Continue or Claim Reward visually resembling a functional app button, which triggers an ad redirection upon click.

User Impact: Deceptive UI ads reduce user trust by inducing unintended interactions and increasing cognitive load during use.

3.2.2 [T2] *Functionality-Interrupting Ads.* This category represents a violation of task continuity. These ads occur when ad-triggering logic co-opts the user’s legitimate interaction pathways within s_{task} , turning functional navigation into involuntary ad engagement.

(T2.1) Hijacked Functional Controls

- *Description.* Within a task state (s_{task}), interacting with a functional control v (i.e., $\Phi(v) \in \Phi_{app}$) should normally lead to a functional state transition. Instead, the interaction redirects the user to an ad state (s_{ad}), hijacking the intended workflow.

$$\exists v \in V(s_{task}) : (\Phi(v) \in \Phi_{app}) \wedge (\delta(s_{task}, \text{interact}(v)) \in S_{ad})$$

- *Example.* Pressing “Next” in a tutorial launches a full-screen ad instead of advancing the step.

(T2.2) Interaction-triggered Activations

- *Description.* Ads are triggered by generic, non-targeted gestures $e_{gen} \in \mathcal{E}$ (e.g., scrolling, swiping, or tapping white space) that lack any explicit intent to invoke an ad. This mechanism exploits the high frequency of these navigation gestures to maximize accidental ad impressions.

$$\delta(s_{task}, e_{gen}) \in S_{ad}, \quad \text{where } e_{gen} \notin \{e_{intent}, e_{close}\}$$

- *Example.* While a user is scrolling through a news feed (e_{gen}), the app misinterprets the scrolling motion as an ad-click, popping up a floating ad and disrupting the reading flow.

User Impact: Functionality-interrupting ads cause cognitive fragmentation by coupling ad triggers with core task events. This prevents users from predicting or avoiding ad exposure, increasing cognitive load and leading to frustration and task abandonment.

3.2.3 [T3] *Multi-Step Exit Ads*. This type refers to ads that require multiple interactions before the user can return to the intended task, exploiting user effort to increase engagement.

- *Description*. Let $Path(s, s')$ denote the interaction path from state s to s' , whose intermediate states may range over $S_{ad} \cup S_{target}$, and $|Path|$ be the number of user interactions along this path. An ad state s_{ad} is classified as intrusive if the shortest path to resume the intended task requires more than one interaction:

$$\min_{s_{task} \in S_{task}} |Path(s_{ad}, s_{task})| > 1$$

- *Example*. An ad demands clicking through multiple confirmation screens to close.

User Impact: Multi-step exit ads artificially inflate the interaction cost of ad dismissal. These ads exploit user impatience to force accidental engagement, significantly obstructing the return to the intended task and degrading the efficiency of the primary workflow.

3.2.4 [T4] *Aggressive Redirection Ads*. This category physically removes users from their intended context, without explicit consent.

(T4.1) Automatic Redirection

- *Description*. The transition to an ad target state s_{target} occurs automatically while the user is in the ad state s_{ad} , triggered merely by the passage of time (e_{wait}), rather than a voluntary interaction.

$$\delta(s_{ad}, e_{wait}) \in S_{target}$$

- *Example*. A full-screen ad immediately redirects to an app store upon loading, without the user touching the screen.

(T4.2) Sensor-based Redirection

- *Description*. The transition is triggered by environmental sensor events e_{sensor} whose motion magnitude $mag(e_{sensor})$ falls below the threshold of intentionality τ_{intent} . This effectively converts ambient physical movement into an involuntary redirection command.

$$\delta(s_{ad}, e_{sensor}) \in S_{target}, \quad \text{where } mag(e_{sensor}) < \tau_{intent}$$

- *Example*. An ad redirects the user to an external website simply because the user shifted their phone slightly while walking (triggering the gyroscope/accelerometer), effectively “shaking” the user out of the app.

User Impact: Aggressive redirection ads bypass users’ agency by initiating transitions without interactive consent, resulting in involuntary displacement from the original context to ad content and loss of navigational control.

3.2.5 [T5] *Excessive Frequent Ads*. This type repeatedly interrupts the user flow by presenting multiple ads in close temporal proximity, preventing users from resuming normal application tasks.

- *Description*. Let $\pi = \langle s_1, \dots, s_n \rangle$ be a UI interaction trace. This type is identified if *either* of the following conditions holds:

- (i) **Ad Density**: The proportion of ad states encountered in the interaction trace exceeds a predefined threshold θ .

$$\frac{|\{s_i \in \pi \mid s_i \in S_{ad}\}|}{|\pi|} > \theta$$

- **(ii) Sequential Back-to-Back Ads:** The user encounters a succession of at least κ ad states, such that after dismissing one ad, only a negligible number of functional interactions (at most g) occur before the next ad appears.

$$\exists s_{i_1}, \dots, s_{i_\kappa} \in \pi \cap S_{ad} (i_1 < \dots < i_\kappa) : \forall j \in [1, \kappa-1], i_{j+1} - i_j \leq g + 1$$

- *Example.* An ad pops up every time the user completes a simple task.

User Impact: Excessive ad frequency causes sustained disruption, increases cognitive fatigue, and significantly degrades user experience by obstructing task completion.

Response to RQ1: We propose a taxonomy of mobile intrusive advertising consisting of five categories: Deceptive UI Ads, Functionality-Interrupting Ads, Multi-step Exit Ads, Aggressive Redirection Ads, and Excessive Frequent Ads. Each category is formally specified with symbolic definitions and illustrative examples, capturing distinct intrusive patterns and their impacts on user experience.

4 Methodology

To systematically investigate intrusive advertising behaviors in the Android ecosystem, we first construct two complementary datasets for large-scale testing and regulatory assessment. Given the collected apps, we conduct an online dynamic exploration to collect ad-relevant runtime data, followed by offline multi-modal analysis to identify and characterize intrusive advertising behaviors, as illustrated in Figure 2.

4.1 Applications Collection

To systematically investigate intrusive advertising behaviors across different contexts of apps, we curated two complementary datasets.

(1) *Broad-Coverage Apps Dataset.* This dataset consists of **6,447** Android apps collected from multiple markets, including **4,411** apps from AndroZoo [4] and **2,036** apps from AppChina [5], a popular Chinese app market. To reflect contemporary Ad SDK deployments and monetization practices, we restricted our analysis to applications released or updated after January 2024.

(2) *Policy-relevant Apps Dataset.* To examine intrusive advertising behaviors associated with regulatory risk, we constructed a policy-relevant dataset based on public notifications released by MIIT [33]. As the notifications do not adopt a unified terminology for intrusive advertising, we selected apps whose reported problems explicitly involved pop-up or redirection issues that are highly indicative of intrusive interaction patterns. The notifications provide app names, versions, and brief issue descriptions, but lack package identifiers or other metadata. We therefore searched for the reported app names and corresponding versions across third-party Android markets, resulting in a dataset of **55** applications aligned with the notification records.

4.2 Online Dynamic Exploration

This phase gathers ad-related transitions and runtime contexts within the app. Specifically, it takes the APK as input, then collects multi-modal runtime traces while building a dynamic UI Transition Graph (UTG) with ad-related execution contexts, guided by an ad-oriented exploration strategy and extended actions injection, providing the necessary data foundation for the following analysis.

4.2.1 Contextual Stimulus Generation. User interactions constitute the primary pathway through which ads are triggered. Moreover, many ad exposures appear only under specific interaction sequences, rather than isolated UI events. To expose a broader range of ad-triggering conditions, this module synthesizes executable interactions based on the current runtime context and active

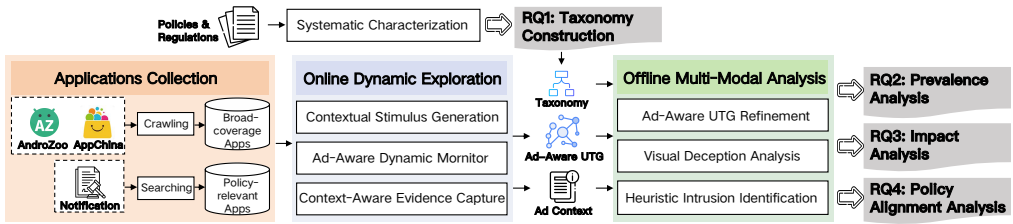


Fig. 2. Overview of our methodology.

exploration strategy. **1) Systematic UI Traversal.** As ad exposures often require sustained or compound user engagement, this module performs a recursive traversal of interactive UI widgets, starting from the app’s entry point. Widgets are categorized by interaction types (e.g., clickable and scrollable), and corresponding primitive actions (e.g., click and scroll) are applied to traverse the functional topology. This ensures baseline coverage of the UI states. **2) Context-Aware Action Augmentation.** However, primitive actions alone are insufficient for intrusive ad detection. Our design considers the dual objective: (1) *maximizing the ad exposure*, and (2) *collecting concrete execution evidence demonstrating how ads interfere with normal user operations*. To achieve these objectives, we extend the original action space by introducing action augmentation. We design specific interaction patterns to emulate realistic user behaviors:

- **Continuous Interaction Gestures:** We introduce interaction patterns commonly observed during content consumption, such as *continuous scrolling*, to expose ads that interrupt ongoing user engagement, including lazily loaded or conditionally embedded ads.
- **Composite Action Chains:** We construct coherent interaction sequences (e.g., $e_{wait} \rightarrow e_{click} \rightarrow e_{back}$) to validate advertising intrusiveness. The waiting phase reveals unsolicited launches or aggressive redirections, while the back action evaluates whether ads can be dismissed without obstructing normal navigation.
- **Sensor-Based Stimuli:** We incorporate lightweight sensor-based actions, such as simulating device shaking under users’ normal operation (e.g., walking), to uncover ads triggered by incidental physical movements rather than explicit user intent.

4.2.2 Ad-Aware Dynamic Monitor. To identify ad-related behaviors during execution, this module is designed to continuously observe runtime signals indicative of ad presence. This module operates alongside interaction execution and provides the necessary runtime context for ad-oriented exploration and intrusiveness assessment. Specifically, it combines runtime API instrumentation and UI-level heuristics to detect ads across different execution layers. We first instrument selected runtime interfaces to capture ad-related events, such as WebView loading and navigation triggers commonly associated with in-app advertising. At the UI layout, we inspect view hierarchies and layout attributes to identify ad-related controls and presentation patterns. Building on the standard UTG abstraction, we further enrich each explored UI state with ad-related semantic annotations. For each state, this module records whether ad elements are present and associates them with view-level characteristics. These annotations enable precise associations among UI states, user interactions, and ad-triggering events, facilitating downstream analysis of ad-triggering conditions. By undertaking more granular annotation, the module provides a practical foundation for targeted analysis of intrusive advertising behaviors.

4.2.3 Context-Aware Evidence Capture. To support the reliable assessment of intrusive ads, this module is designed to record critical execution evidence associated with ad-related interactions. For each captured step, we collect a multi-modal execution record that includes: **(1) Fine-grained UI**

structure from view hierarchies, enabling the identification of deceptive UI elements. **(2) Dynamic execution signals** including activity transitions, embedded landing site URLs, API traces, event sequences, and temporal patterns. **(3) Intermediate and transient UI states** that are crucial for detecting intrusive behaviors, such as pop-up layers, delayed dismiss screens, or transitional overlays preceding forced navigation.

4.3 Offline Multi-Modal Analysis

This phase aggregates and refines the collected evidence to construct an enhanced, ad-aware UTG by filtering noise and supplementing fine-grained metadata. Through joint reasoning over visual cues, structural transitions, and behavioral signals, it maps app activities to taxonomy-defined patterns for explainable intrusiveness inference.

4.3.1 Ad-Aware UTG Refinement. This module distills the raw UTG into an ad-aware UTG by synthesizing recorded view hierarchies, interaction contexts, and ad-related annotations. This refinement step serves two purposes: *(1) Filtering out UI elements and transitions that are not related to advertising.* *(2) Supplementing ad-relevant semantics that may be ambiguous or weakly indicated during runtime.* To achieve this, we apply empirically derived false-positive keyword lists and conduct a granular examination of key view attributes (e.g., `resource_id` and `content_description`) to correct misclassified ad nodes and strengthen ad-related associations. The resulting ad-aware UTG preserves the original transition structure while providing a cleaner and more semantically grounded foundation for downstream analysis.

4.3.2 Visual Deception Analysis. To detect Deceptive UI Ads, we analyze screenshots associated with corresponding UI states through a two-stage pipeline.

Stage1: Ad Region Localization. We first localize ad regions within UI screenshots of the ad state. Due to the scarcity of labeled data for ad region localization, we employ a data augmentation strategy by synthetically embedding advertising assets into screenshots of real apps. The augmentation is guided by domain knowledge and excludes unrealistic transformations (e.g., extreme transparency or arbitrary rotations) that rarely occur in practice. We consider four common ad formats observed in mobile apps: full-screen ads, pop-up ads, banner ads, and native ads. The augmentation strategy is tailored to each ad format according to its characteristic placement patterns. For instance, full-screen ads receive minimal scaling to preserve their dominant visual presence, while banner ads are positioned at the top or bottom of the screen to reflect their typical layout. Across all formats, opacity is adjusted within a high-visibility range to maintain realism. The final dataset comprises over 30,000 augmented images. Using this dataset, we train a YOLOv8-based object-detection model [36] to identify ad regions. All subsequent deception analysis is restricted to the detected ad regions. UI elements outside these regions are ignored to avoid conflating normal app controls with ad-related interactions.

Stage2: Intra-region Deception Reasoning. Given a localized ad region, we further analyze whether it exhibits visually deceptive exit or misleading interaction designs.

- **Exit Control Assessment.** We examine whether the ad provides a clear and usable dismissal mechanism. A YOLOv8-based detector [36] is trained to distinguish clear exit controls from obscured exits. The training dataset is constructed through a combination of real ad screenshots and controlled synthetic augmentation to model realistic exit design variations. For the clear-exit dataset, we collect diverse exit control styles (e.g., “x”, “close” with different shapes and contrasts) and place them into screenshots using partial augmentation, including moderate position variation and scaling, for preserving visual clarity. This process yields approximately 15,000 annotated samples representing clearly visible exit controls. For the obscured-exit dataset, we simulate deceptive design patterns by applying visual obfuscation strategies to exit controls before embedding them into screenshots. We

apply transformations to ensure the datasets contain obscured configurations, including multiple exit controls within the same ad region, visually blurred exits, and overlapping exits.

- **Misleading Affordance Detection.** We identify deceptive interaction cues based on a functional consistency principle of ad interfaces. In standard ad design, legitimate interactive controls within an ad region are typically restricted to exit controls (e.g., close/skip) and ad-entry controls (leading to ad redirection). We apply UIED [34] to extract individual interactive and visual components within the ad region. Components that exhibit visually interactive affordances but are not semantically associated with either dismissal or ad-entry functions are classified as misleading affordances. Such elements create a mismatch between perceived interactivity and actual functional intent, indicating deceptive interface design.

Detect [T1] Deceptive UI Ads. An ad state is classified as T1 if, within a detected ad region, any of the following holds: (1) no clear exit control is detected (T1.1); (2) an obscured exit control is detected (T1.2); (3) multiple exit-like controls are detected within the same ad region, as a static approximation of exit ambiguity (T1.3); or (4) interactive components are present whose visual semantics correspond to neither dismissal nor ad entry (T1.4).

4.3.3 Heuristic Intrusion Identification. While Deceptive UI Ads manifest primarily through visual manipulation, other intrusive ad types emerge from interaction dynamics and execution context. This subsection focuses on identifying T2-5 in our taxonomy by operationalizing the formal definitions introduced in Section 3 into concrete graph traversal rules.

T2: Functionality-Interrupting Ads. These behaviors hijack or disrupt the user’s intended workflow. We identify them by analyzing action–state causality in the UTG.

- **Hijacked Functional Controls:** An intentional interaction e_{intent} on a functional control, expected to yield a functional transition, instead satisfies $\delta(s_{task}, e_{intent}) \in S_{ad}$, indicating that the control is repurposed as an ad trigger.
- **Interaction-Triggered Activations:** A generic navigation gesture satisfies $\delta(s_{task}, e_{gen}) \in S_{ad}$, i.e., ads are activated by target-less interactions carrying no ad intent.

T3: Multi-Step Exit Ads. An ad is classified as T3 if $\min_{s_{task} \in S_{task}} |Path(s_{ad}, s_{task})| > 1$ on the ad-aware UTG. In practice, we conservatively test whether a single-step exit succeeds: if a back-key event (e_{close}) from a state in $S_{ad} \cup S_{target}$ leads to another ad or target state, rather than S_{task} , the ad requires at least two interactions to resume the user’s task.

T4: Aggressive Redirection Ads. These patterns force users out of the host application without explicit consent, manifested as unauthorized foreground package transitions.

- **Automatic Redirection:** We check for transitions $\delta(s_{ad}, e_{wait}) \in S_{target}$, where e_{wait} is captured via *WaitEvent*, which records state changes occurring in the absence of touch input.
- **Sensor-Based Redirection:** We identify transitions $\delta(s_{ad}, e_{sensor}) \in S_{target}$ by injecting controlled motion signals via ADB [17]. The injected events are parameterized by acceleration magnitude, angular displacement, and duration. Following the official Shake-to-Interact Advertising Guidelines [27], a redirection is flagged if $mag(e_{sensor}) < \tau_{intent}$, where $mag(e_{sensor})$ denotes the motion magnitude of the injected sensor event.

T5: Excessive Frequent Ads. Since banner ads remain embedded in the page without seizing the interaction focus, banner-only states are treated as functional states and excluded from this analysis. A trace is flagged if either of the following operational criteria is satisfied:

- **Ad Density.** The proportion of ad states along an extracted interaction trace π exceeds θ : $|\{s_i \in \pi \mid s_i \in S_{ad}\}| / |\pi| > \theta$, where $\theta = 0.4$ is chosen empirically.

- *Back-to-back Ad Sequences.* At least $\kappa = 3$ distinct ad states occur with at most $g = 2$ intervening functional states between consecutive ads. UI fingerprints (activity, resource identifiers, and view texts) distinguish ad states to avoid duplicate counting.

5 Results and Findings

We apply our approach to datasets comprising real-world applications. In this section, we present the overall results of the mobile intrusive ads and break down each type with further analysis.

5.1 Overall Prevalence and Distribution

5.1.1 Overall Prevalence. The number of apps with intrusive ads is not negligible, and they are more prevalent in a specific marketplace. Table 1 summarizes the detection results. Across the dataset, 7.46% of APKs contain intrusive ads. The proportion is higher in the AppChina marketplace (12.57%) than in AndroZoo (5.10%). This indicates that intrusive ads genuinely exist in Android app markets, with AppChina exhibiting a higher prevalence.

5.1.2 Distribution of Different Intrusive Types. We categorize the detected intrusive ads using our taxonomy to measure the prevalence of each type. Since an application may exhibit multiple intrusive behaviors simultaneously, we report the percentage of applications per type, meaning the totals do not sum to 100%.

As illustrated in Figure 3, Functionality-interrupting Ads (T2) emerge as the most dominant type across both marketplaces, manifesting in 76.89% and 89.06% of the detected applications in AndroZoo and AppChina, respectively. This finding suggests that intrusive advertising strategies increasingly pivot toward disrupting users' task flows rather than merely relying on visually aggressive interfaces. Notably, such behaviors are often deeply embedded within legitimate application workflows, thereby increasing the difficulty for users to distinguish them from standard interactions.

The remaining intrusive types exhibit both shared commonalities and marketplace-specific discrepancies. Deceptive UI Ads (T1) appear at nearly identical rates in both ecosystems (32.44% vs. 31.64%), indicating that conventional visual intrusion remains a pervasive strategy. Conversely, Excessive Frequent Ads (T5) demonstrate nearly a twofold increase in prevalence within AndroZoo compared to AppChina (43.11% vs. 23.44%), implying divergent strategic preferences in advertising deployment across these marketplaces. Finally, while T3 remains consistently rare in both environments, T4 is uniquely observed in AppChina (1.17%). Although minimal, this presence underscores that certain intrusive behaviors may be idiosyncratic to specific application ecosystems.

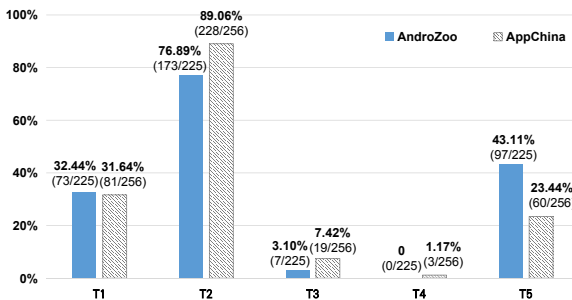


Fig. 3. Distribution of different intrusive types. Note that a single app can have multiple intrusive ad types.

Table 1. Overall Detection Results of Intrusive Ads in Android Apps collected from AndroZoo and AppChina.

Metric	# APKs Analyzed	# APKs with Intrusive Ads
AndroZoo	4,411	225 (5.10%)
AppChina	2,036	256 (12.57%)
Total	6,447	481 (7.46%)

5.1.3 Co-occurrence and Combination of Intrusive Types. Intrusive ad types are commonly observed in both standalone and combined forms. Functionality-interrupting Ads (T2) constitute the most

Table 2. Distribution of Ad Formats in Intrusive Ads.

Ad Format	AndroZoo	AppChina
Banner	49.73%	34.51%
Native	7.10%	9.86%
Full-screen	30.60%	37.32%
Pop-up	12.57%	18.31%

Table 3. App-Level Distribution of Content Risk Categories in Intrusive Ads.

Risky Content	AndroZoo	AppChina
R1	2.67%	2.34%
R2	4.00%	14.84%
R3	10.22%	0%

common standalone intrusive type. Among applications exhibiting multiple intrusive types, the most frequent combinations are T2+T5, followed by T1+T2 and T1+T2+T5. These recurring combinations further highlight the central role of T2, which frequently layered with repeated ad exposure (T5) and, in some cases, visually aggressive interface designs (T1). The coexistence of these intrusive behaviors may contribute to a more intrusive advertising experience.

Response to RQ2: Our study reveals a 7.46% prevalence of mobile intrusive ads. Functionality-interrupting Ads (T2) is the most common standalone intrusive type, while T2+T5 and T1+T2 are the two most frequently observed multi-type combinations.

5.2 Impact Analysis

5.2.1 Manifestation Across Ad Formats. Mobile ads are commonly implemented through various formats, including *banner ads*, *full-screen ads*, *native ads*, and *pop-up ads*. Table 2 summarizes the observed ad formats in the broad-coverage apps dataset. Banner and full-screen ads account for the majority of intrusive ads across both sources. While banner ads are more prevalent in AndroZoo (49.73%), full-screen and pop-up ads appear more frequently in AppChina, suggesting that different markets tend to adopt distinct ad integration and monetization strategies.

Although banner and native ads are generally considered less disruptive because they are designed to coexist with application content, our observations show that they can still become intrusive through inappropriate implementation. Specifically, we frequently observe three recurring implementation patterns: (1) *disproportionate advertisement size*, where ads occupy an excessive portion of the interface or dominate the content area; (2) *layout obstruction*, where ads overlap critical interaction regions or obscure functional interface elements; and (3) *ineffective dismissal mechanisms*, where ads provide no visible or functional means of dismissal. These implementation patterns transform ads that are otherwise intended to be persistent or content-integrated into intrusive user experiences.

These observations suggest that ad format alone is a poor predictor of intrusiveness. Instead, intrusive behaviors arise primarily from how ads interact with application layouts and user workflows. In other words, implementation choices rather than display formats largely determine whether an advertisement becomes intrusive.

5.2.2 Content Risk Analysis of Intrusive Ads. Beyond interface-level interference, intrusive advertisements may also expose users to content-related risks. We manually analyze the landing pages and promoted content of intrusive ads and identify several recurring risk categories.

- **(R1) Inappropriate content**, including sexually explicit or otherwise unsuitable materials that may negatively affect users, especially minors.
- **(R2) Untrustworthy content**, including ads that employ fraudulent interaction flows, such as fabricated approval status or unsolicited requests for personal information, or that lead to low-trust destinations such as phishing domains and sites with security misconfigurations.
- **(R3) Regulation-sensitive content**, including religious materials and VPN services, which may violate local regulations or raise compliance concerns.

Table 4. Overall Results of Intrusive Ads in Notification-Reported Apps.

Total APKs	#APKs with ads	#APKs with intrusive ads	#T1	#T2	#T3	#T4	#T5
55	14 (25.5%)	14 (25.5%)	2	12	3	1	6

A notable regional contrast emerges in the composition of these risks. As shown in Table 3, the two sources exhibit comparable overall risk prevalence but markedly different profiles: risky ads in AppChina are dominated by untrustworthy content (R2), whereas those in AndroZoo are characterized by regulation-sensitive content (R3), which is entirely absent from AppChina. This observation suggests that interaction-level intrusiveness and content-related risks capture different aspects of advertising harm. Ads may be highly intrusive without promoting risky content, while content carrying security or regulatory concerns may not necessarily rely on intrusive interaction mechanisms. These findings highlight the importance of evaluating intrusive advertising beyond interface disruption alone.

5.2.3 Regulatory and Compliance Implications. To examine whether intrusive advertising behaviors also appear in applications associated with real-world policy concerns, we analyzed the policy-relevant dataset derived from MIIT notification records. Among the 55 policy-related apps, 14 contain ads, and all ad-enabled applications exhibit at least one intrusive advertising behavior defined in our taxonomy, as shown in Table 4. The result demonstrates that they encompass all types of intrusive patterns we define in the taxonomy. Although the notification records do not explicitly classify these applications as intrusive-ad cases, the reported issues frequently involve pop-up windows, unexpected redirections, and other interaction patterns closely related to intrusive advertising. The widespread presence of intrusive ads in this dataset therefore provides additional evidence that the behaviors captured by our taxonomy are not limited to isolated cases, but also emerge in applications associated with real-world policy and user-protection concerns.

This observation suggests that intrusive advertising frequently co-occurs with broader user-experience and compliance-related issues, highlighting its practical relevance beyond controlled measurement settings.

Response to RQ3: *The impact of intrusive advertising extends beyond interface disruption. Our analysis shows that intrusiveness is largely shaped by implementation choices rather than advertisement formats, may coexist with content-related risks, and is frequently observed in policy-relevant applications associated with user-protection concerns.*

6 Policy and Regulation Alignment

6.1 Alignment Mapping

To evaluate the extent to which intrusive advertising behaviors are addressed by existing governance frameworks, we map our taxonomy against five representative policy and regulatory documents, as summarized in Table 5. We consider two categories of governance instruments: **(1) Platform/industry standards**, including Google Play Ads Policies [16], the Better Ads Standards [9] and the Acceptable Ads [3]. **(2) Regional regulations**, encompassing China’s Internet Advertising Management Measures [29], and Guidelines on the Protection of Personal Rights in “Shake-to-Trigger” Advertising [27], as well as Korea’s Fair Trade Commission (FTC) Review Guidelines [21]. For each intrusive advertising pattern, we assess the degree to which it is addressed by a given framework and assign one of four restriction levels: *Explicitly Restricted* (ER), *Partially Specified* (PS), *Implicitly Discouraged* (ID), and *Not Addressed* (NA). This mapping enables a systematic comparison between observed intrusive advertising behaviors and existing policy coverage.

Table 5. Mapping of Restriction Levels for Specific Ad Patterns Across Major Regulatory Frameworks

Intrusive Ad Patterns		Platform / Industry Standards			Regional Government Regulations	
Main Type	Pattern	Google Play	Better Ads	Acceptable Ads	China	Korea
T1	T1.1	ER	PS	PS	ER	ID
	T1.2	ER	ID	NA	ER	ID
	T1.3	ID	NA	NA	ID	ID
	T1.4	ER	NA	NA	ER	NA
T2	T2.1	ER	ER	PS	ID	NA
	T2.2	PS	PS	NA	PS	NA
T3	/	PS	NA	NA	ER	NA
T4	T4.1	PS	PS	ID	PS	NA
	T4.2	NA	NA	NA	ER	NA
T5	/	ER	ID	ER	ER	NA

¹ **ER** (Explicitly Restricted): The behavior is explicitly and directly prohibited.

² **PS** (Partially Specified): The behavior is directly targeted but criteria are incomplete or conditional.

³ **ID** (Implicitly Discouraged): The behavior is only indirectly covered through broad principles.

⁴ **NA** (Not Addressed): The behavior is not mentioned in the policies/regulations.

6.2 Comparison of Policies

6.2.1 Cross-Framework Differences. Table 5 shows that existing policies do not address intrusive advertising behaviors uniformly. While some behaviors are explicitly prohibited across multiple documents, others are only partially specified or remain entirely unaddressed. The following comparison highlights how different governance instruments prioritize distinct aspects of intrusive advertising. Google Play and China’s regulations address the widest range of intrusive advertising patterns in our taxonomy, although their emphasis differs. Google Play more frequently adopts explicit prohibitions for common deceptive and disruptive behaviors, particularly T1 and T2 patterns, and supplements these restrictions with illustrative examples. In contrast, Chinese regulations place greater emphasis on implementation details related to user control, such as one-step dismissal, authentic close controls, and repeated pop-up restrictions, while also uniquely addressing sensor-triggered redirection (T4.2). The Better Ads Standards mainly target disruptive advertising formats, with explicit restrictions focusing on interstitial experiences and functionality interruption. By contrast, the Acceptable Ads Standard emphasizes format and frequency requirements, resulting in ER-level only for excessive advertising frequency (T5) while leaving many interaction-level behaviors insufficiently specified. Korea’s FTC Review Guidelines adopt a principle-based approach, addressing several deceptive practices indirectly but providing limited operational guidance for specific implementations.

Consequently, these differences extend beyond regulatory scope to the granularity of specification. While certain frameworks define concrete implementation requirements, others rely on broad principles or illustrative examples. Consequently, intrusive implementations that undermine practical dismissibility or user autonomy may remain formally compliant despite imposing substantial disruption on users.

6.2.2 Empirical Implications. Comparing policy coverage with empirical observations reveals several discrepancies between policy design and market reality. Although T2 represents the most prevalent violation type in AndroZoo (76.89%) and AppChina (89.06%), its subtype T2.2, which interrupts users through generic or unintended interactions, is only partially covered by existing advertising policies and is not explicitly prohibited by any of the surveyed frameworks. This discrepancy indicates that some of the most frequently observed intrusive behaviors remain insufficiently addressed by current policy specifications. At the other end of the spectrum, aggressive

redirection advertisements (T4) are rarely observed and appear only in AppChina (1.17%), suggesting that certain intrusive behaviors may be marketplace-specific and are therefore not fully reflected by existing policy frameworks.

Overall, these unequal distributions across advertising categories suggest that intrusive advertising strategies vary systematically across application marketplaces.

Response to RQ4: Existing policies provide uneven coverage of intrusive advertising implementations, leaving several frequently observed intrusive behaviors insufficiently specified in current policy frameworks.

7 Discussion

7.1 Detection Accuracy

To evaluate the reliability of our automated detection pipeline, we conducted a manual validation study on applications identified as containing intrusive ads. Among those applications, we employed proportionate stratified sampling and manually inspected 80 applications (38 from AndroZoo and 42 from AppChina). This sample size provides approximately a 95% confidence level with a 10% margin of error for estimating the overall agreement rate. Within each source, the sampled applications were further selected to ensure that all intrusive categories were represented.

Validation is conducted at the application and category level. For each sampled application, we verify whether each reported intrusive advertisement category is supported by the collected execution traces, screenshots, and UI transitions. A reported category is considered validated when at least one corresponding intrusive behavior is confirmed through manual inspection, while any observable but unreported behaviors are recorded as missed detections.

Table 6 summarizes the validation results across the five categories. The detection results show high agreement with manual inspection, with an overall validation rate of 90% across 214 reported detections and 22 missed cases in total. T1 and T4 achieve full validation, while most unconfirmed reports concentrate in T3 and T5, which we analyze below.

7.2 Error Analysis

The observed validation disagreements mainly arise from three intrinsic characteristics of intrusive ads. **First, intrusive ads exhibit increasing visual ambiguity.** Native-style layouts, transparency, animation, and irregular interface designs blur the boundary between advertisements and legitimate application content, making visually intrusive patterns difficult to distinguish from normal UI elements. This primarily contributes to the missed cases in T1 and also affects transition-based behaviors (T2 and T4), whose identification relies on correctly recognizing ad states before reasoning about user interactions. **Second, similar interface transitions may arise from different underlying causes.** An ad appearing immediately after a legitimate user action may either be causally triggered by that action or occur independently due to runtime ad delivery, making the resulting interruption semantically ambiguous. Conversely, ads that are not displayed during a particular execution leave the corresponding interrupting behaviors unobserved. These factors account for the remaining disagreements in T2. **Third, intrusive ads exhibit substantial interaction variability.** Behaviors such as multi-layer ads, browser-mediated interactions, external popups, and cross-context redirections alter the intended interaction sequence by introducing additional interface and navigation contexts. Consequently, user actions such as dismissal, back navigation, or repeated exposure may no longer produce their expected interaction outcomes, while intermediate interaction stages may become difficult to observe. These characteristics account for most of the remaining disagreements in T3, T4, and T5. Overall, these observations suggest that

Table 6. Manual validation results at the application and category level.

Type	Reported	Validated	Validation Rate	Missed
T1	51	51	100%	15
T2	77	74	96%	3
T3	25	16	64%	1
T4	2	2	100%	2
T5	59	49	83%	1
Overall	214	192	90%	22

the fundamental challenges lie not only in distinguishing increasingly complex ads from legitimate application interfaces, but also in characterizing the intricate interaction patterns such as layered navigation and cross-context ad flows.

7.3 Threats to Validity

Internal validity. Our analysis relies on dynamic UI exploration to trigger and detect intrusive behaviors potentially missing non-human interaction behaviors and causing false negatives. To mitigate this, we periodically reset the emulator environment to reduce likelihood of being flagged.

External validity. Although we conducted an empirical study on a large-scale dataset including over 6000 apps, it may not fully capture the diversity of real-world deployment conditions since there are no existing standard benchmark datasets for intrusive advertisements of mobile applications, which are collected by other studies. To mitigate this, we include apps from multiple sources (AndroZoo and AppChina), covering different app distributions.

Construct validity. Our taxonomy was manually constructed, which may pose potential threats to validity. To mitigate subjectivity, we grounded our classification in multiple sources, including official advertising review guidelines from the market, industry, and regulatory authorities, supplemented by iterative refinements from real-world app observations. Such alignment with established industry standards minimizes individual bias and ensures classification consistency.

8 Related Work

8.1 Intrusive Advertising on Web Platform

Several works have examined intrusive advertising practices on web platform. Yan et al. proposed AdHere [35] to detect intrusive ads that violate Better Ads Standards. They discussed 4 unacceptable desktop ads and 8 unacceptable mobile ads based on the Better Ads, and detected violating ads on 5,540 mobile websites and 4,601 desktop websites in the Alexa Top 1 Million websites. Zafar et al. [37] conducted a comprehensive assessment of web advertising practices permitted under the Acceptable Ads Standards. By developing a crawler capable of identifying ads and capturing their CSS properties, they analyzed 10,000 popular domains at the intersection of Tranco’s Top 100K and the Acceptable Ads exception list. Despite these domains being explicitly allowed to display “non-intrusive” ads, their study revealed that nearly 10% still contain non-compliant ads on landing pages. They demonstrated that intrusive ads do exist on web platforms and analyzed them by leveraging browser-level instrumentation, DOM structure, and CSS features. However, in-app mobile ads fundamentally differ. In-app ads are rendered via native SDKs rather than HTML/CSS, can appear across activities, overlays, and system surfaces, and are triggered by lifecycle events or sensor inputs. Thus, existing web-centric detection methodologies and standards cannot be directly applied to characterize intrusive behaviors in mobile applications.

8.2 Mobile Ad Fraud and Security

Ad fraud detection has been a major focus in this area. DECAF [23] detected impression fraud on Windows Phone through automated app navigation. FraudDroid [12] pioneered Android ad fraud detection by identifying nine fraud types through dynamic UI exploration. MadDroid [24] further analyzed malicious ad content through network traffic inspection and content analysis, while FraudDetective [20] identified fraudulent ad activities using dynamic testing and attributed most cases to third-party ad libraries. These works build upon fundamental techniques [6, 11, 13, 22, 39], including Android program analysis, automated GUI exploration, dynamic testing, and network analysis, to systematically analyze advertising behaviors for fraud detection.

While these works make significant contributions to detecting fraudulent and malicious mobile ads, they primarily focus on violations that directly threaten advertisers, platforms, or revenue. They assume that harmful behaviors must involve financial gain, deception, or security risks, and rely heavily on detecting anomalies in ad placement, content, or network interactions. In contrast, our work shifts the focus to user-centric intrusiveness, examining how mobile ads harm user experience through UI manipulation and behavioral disruption. We assume all user behaviors are genuine and non-malicious, evaluating intrusiveness from the perspective of legitimate users accomplishing their intended tasks, not adversarial actors attempting to defraud the ad ecosystem.

8.3 Dark Patterns in Mobile Applications

Research on dark patterns has revealed manipulative UI designs that trick users into unintended actions. Kelly and Rubin [19] characterized dark patterns in account-disabling interfaces through content analysis. AidUI [26] proposed an automated framework for recognizing dark patterns in graphical user interfaces, while UIGuard [8] conducted the first systematic study of mobile dark pattern taxonomies and introduced an automated detector for mobile applications. Cai et al. [7] further developed a computer-vision-based detector targeting asymmetric dark UI patterns.

While dark pattern research broadly targets deceptive interface designs, our work focuses specifically on intrusive advertising behaviors and their interaction mechanisms.

9 Conclusion

This paper presents an exploratory study that systematically investigates and models the landscape of intrusive advertising in the mobile ecosystem. We established a formal taxonomy grounded in regulatory policies, platform governance guidelines, industry standards, and documented real-world ad practices from major SDK providers, providing a unified framework for characterizing intrusive advertising behaviors. Based on this taxonomy, we apply our automated pipeline to analyze more than 6,000 real-world mobile applications. Our empirical study reveals that intrusive ads are prevalent and increasingly integrated into legitimate user interactions, with functionality interruption emerging as the dominant intrusive behavior. While intrusive ads appear in both standalone and combined forms, functionality interruption serves as the core component of the most frequently observed multi-type patterns, highlighting its central role in shaping intrusive user experiences. We further identify a paradox of low-intrusiveness, where ostensibly benign ad formats escalate into significant disruptions when deployed without appropriate behavioral constraints. By uncovering these behavioral characteristics and the uneven policy coverage of intrusive advertising implementations, our work provides both a technical foundation for automated auditing and empirical evidence to support more effective governance of mobile advertising.

10 Data Availability

We make the replication package publicly available, which includes all the code and datasets to reproduce our experiments at https://github.com/Noname0519/intrusive_ad_detection as a repository and at <https://doi.org/10.5281/zenodo.21834476> [2] as an archived file.

Acknowledgments

This work was supported by the National Key R&D Program of China (Grant No. 2024YFE0203800), the Beijing-Tianjin-Hebei Natural Science Foundation Cooperation Project (Grant No. 25JJJC0003), the Tianjin Major Science and Technology Special Project (Grant No. 25ZXSFN00140), the Institute of Information & Communications Technology Planning & Evaluation (IITP)-ICT Creative Con-silience Program grant funded by the Korea government (MSIT)(IITP-2026-RS-2020-II201819), the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIT)(RS-2025-02313526 and RS-2026-25489826), Zhongguancun Academy (Grant No. 20240302), and a Korea University Research Grant.

References

- [1] 2024. Mobile Operating System Market Share Worldwide. <https://gs.statcounter.com/os-market-share/mobile/worldwide>.
- [2] 2026. *Code and data*. <https://doi.org/10.5281/zenodo.21834476>
- [3] Acceptable Ads Committee. 2025. Acceptable Ads Standard. <https://acceptableads.com/standard/>.
- [4] Kevin Allix, Tegawendé F. Bissyandé, Jacques Klein, and Yves Le Traon. 2016. AndroZoo: Collecting Millions of Android Apps for the Research Community. In *Proceedings of the 13th International Conference on Mining Software Repositories*. 468–471. doi:10.1145/2901739.2903508
- [5] AppChina. 2025. AppChina. <http://www.appchina.com/>
- [6] Steven Arzt, Siegfried Rasthofer, Christian Fritz, Eric Bodden, Alexandre Bartel, Jacques Klein, Yves Le Traon, Damien Octeau, and Patrick McDaniel. 2014. FlowDroid: Precise Context, Flow, Field, Object-sensitive and Lifecycle-aware Taint Analysis for Android Apps. In *Proceedings of the 35th ACM SIGPLAN Conference on Programming Language Design and Implementation (PLDI)*. 259–269. doi:10.1145/2594291.2594299
- [7] Zhaoxin Cai, Yuhong Nan, Xueqiang Wang, Mengyi Long, Qihua Ou, Min Yang, and Zibin Zheng. 2023. DARPA: Combating Asymmetric Dark UI Patterns on Android with Run-time View Decorator. In *2023 53rd Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN)*. 480–493. doi:10.1109/DSN58367.2023.00052
- [8] Jieshan Chen, Jiamou Sun, Sidong Feng, Zhenchang Xing, Qinghua Lu, Xiwei Xu, and Chunyang Chen. 2023. Unveiling the Tricks: Automated Detection of Dark Patterns in Mobile Applications. In *Proceedings of the 36th Annual ACM Symposium on User Interface Software and Technology (UIST '23)*. Article 114, 20 pages. doi:10.1145/3586183.3606783
- [9] Coalition for Better Ads. 2022. Better Ads Standards. <https://www.betterads.org/standards/>.
- [10] CSJ Developer. 2025. Support Center. <https://www.csjdeveloper.com/en/supportcenter/5465>.
- [11] Anthony Desnos and Androguard Contributors. 2011. Androguard: Reverse Engineering, Malware and Goodware Analysis of Android Applications. <https://github.com/androguard/androguard>.
- [12] Feng Dong, Haoyu Wang, Li Li, Yao Guo, Tegawendé F. Bissyandé, Tianming Liu, Guoai Xu, and Jacques Klein. 2018. FraudDroid: Automated Ad Fraud Detection for Android Apps. In *Proceedings of the 2018 26th ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering*. 257–268. doi:10.1145/3236024.3236045
- [13] Aldo Cortesi et al. 2010. mitmproxy: A Free and Open Source Interactive HTTPS Proxy. <https://mitmproxy.org/>
- [14] Coalition for Better Ads. 2025. The Better Ads. <https://www.betterads.org/standards/>.
- [15] Google. 2025. Examples of common violations. <https://support.google.com/googleplay/android-developer/answer/9857753#zippy=%2Cexamples-of-common-violations>
- [16] Google. 2025. The Google Ads. <https://support.google.com/googleplay/android-developer/answer/12271244?hl=en-GB#FAQ&zippy=%2Cwhere-can-i-show-i%5B%E2%80%A6%5D-the-same-as-an-app-loading-page>
- [17] Google LLC. 2023. *Android Debug Bridge (adb)*. Google. <https://developer.android.com/tools/adb>
- [18] Ziyao He, Syed Fatul Huq, and Sam Malek. 2024. “I Tend to View Ads Almost Like a Pestilence”: On the Accessibility Implications of Mobile Ads for Blind Users. In *2024 IEEE/ACM 46th International Conference on Software Engineering (ICSE)*. 2432–2444. doi:10.1145/3597503.3639228
- [19] Dominique Kelly and Victoria L. Rubin. 2024. Identifying Dark Patterns in User Account Disabling Interfaces: Content Analysis Results. *Social Media + Society* 10, 1 (2024). doi:10.1177/20563051231224269

- [20] Joongyum Kim, Jung-hwan Park, and Soeul Son. 2021. The Abuser Inside Apps: Finding the Culprit Committing Mobile Ad Fraud. In *Proceedings 2021 Network and Distributed System Security Symposium* (2021). doi:10.14722/ndss.2021.23161
- [21] Korean Fair Trade Commission. 2021. Guidelines for the Review of Internet Advertising. <https://www.law.go.kr/%ED%96%89%EC%A0%95%EA%B7%9C%EC%B9%99/%EC%9D%B8%ED%84%B0%EB%84%B7%EA%B4%91%EA%B3%A0%EC%97%90%EA%B4%80%ED%95%9C%EC%8B%AC%EC%82%AC%EC%A7%80%EC%B9%A8>
- [22] Yuanchun Li, Ziyue Yang, Yao Guo, and Xiangqun Chen. [n. d.]. Droidbot: a lightweight ui-guided test input generator for android. In *2017 IEEE/ACM 39th International Conference on Software Engineering Companion (ICSE-C)*. 23–26. doi:10.1109/ICSE-C.2017.8
- [23] Bin Liu, Suman Nath, Ramesh Govindan, and Jie Liu. 2014. DECAF: Detecting and characterizing ad fraud in mobile apps. In *USENIX symposium on networked systems design and implementation (NSDI 14)*. 57–70.
- [24] Tianming Liu, Haoyu Wang, Li Li, Xiapu Luo, Feng Dong, Yao Guo, Liu Wang, Tegawendé Bissyandé, and Jacques Klein. 2020. MadDroid: Characterizing and Detecting Devious Ad Contents for Android Apps. In *Proceedings of The Web Conference 2020* (Taipei, Taiwan) (WWW '20). Association for Computing Machinery, New York, NY, USA, 1715–1726. doi:10.1145/3366423.3380242
- [25] Shang Ma, Chaoran Chen, Shao Yang, Shifu Hou, Toby Jia-Jun Li, Xusheng Xiao, Tao Xie, and Yanfang Ye. 2025. Careful About What App Promotion Ads Recommend! Detecting and Explaining Malware Promotion via App Promotion Graph. In *Proceedings of the 2025 Network and Distributed System Security Symposium (NDSS)*. doi:10.14722/ndss.2025.230051
- [26] S. M. Hasan Mansur, Sabiha Salma, Damilola Awofisayo, and Kevin Moran. 2023. AidUI: Toward Automated Recognition of Dark Patterns in User Interfaces. In *Proceedings of the 45th International Conference on Software Engineering (ICSE)*. IEEE/ACM, 1958–1970. doi:10.1109/ICSE48619.2023.00166
- [27] National Technical Committee 260 on Cybersecurity of Standardization Administration of China (SAC/TC260). 2025. Shake-to-Open Advertising Personal Rights Protection Specification Guidelines (Draft for Comments V1.0-202501). Cybersecurity Standards Practice Guide. <https://www.tc260.org.cn/upload/2025-01-23/1737601991466023444.pdf>
- [28] Soeul Son, Daehyeok Kim, and Vitaly Shmatikov. 2016. What Mobile Ads Know About Mobile Users. In *Proceedings of the 2016 Network and Distributed System Security Symposium (NDSS)*. The Internet Society. doi:10.14722/ndss.2016.23407
- [29] State Administration for Market Regulation. 2023. Internet Advertising Management Measures (China). https://www.samr.gov.cn/zw/zfxgk/fdzdgknr/fgs/art/2023/art_d93a579afd45413e8576e4623fab348f.html
- [30] Keen Sung, JianYi Huang, Mark D Corner, and Brian N Levine. 2020. Re-identification of mobile devices using real-time bidding advertising networks. In *Proceedings of the 26th Annual International Conference on Mobile Computing and Networking*. 1–13. doi:10.1145/3372224.3419205
- [31] The Insight Partners. 2025. Cross Platform and Mobile Advertising Market Overview, Growth, Trends, Analysis, Research Report (2025-2031). <https://www.theinsightpartners.com/reports/cross-platform-and-mobile-advertising-market>
- [32] Wei Wang, Linlin Yang, and Qian Zhang. 2015. Privacy preservation in location-based advertising: A contract-based approach. *Computer Networks* 93 (2015), 213–224. doi:10.1016/j.comnet.2015.10.020
- [33] Wikipedia. 2026. Ministry of Industry and Information Technology (MIIT). https://en.wikipedia.org/wiki/Ministry_of_Industry_and_Information_Technology.
- [34] Mulong Xie, Sidong Feng, Zhenchang Xing, Jieshan Chen, and Chunyang Chen. 2020. UIED: a hybrid tool for GUI element detection (ESEC/FSE 2020). New York, NY, USA, 1655–1659. doi:10.1145/3368089.3417940
- [35] Yutian Yan, Yunhui Zheng, Xinyue Liu, Nenad Medvidovic, and Weihang Wang. 2023. AdHere: Automated detection and repair of intrusive ads. In *2023 IEEE/ACM 45th International Conference on Software Engineering (ICSE)*. 486–498. doi:10.1109/ICSE48619.2023.00051
- [36] YOLO team. 2025. YOLOv8. <https://docs.ultralytics.com/models/yolov8/>
- [37] Ahsan Zafar and Anupam Das. 2025. Assessing Compliance in Digital Advertising: A Deep Dive into Acceptable Ads Standards. In *Proceedings of the ACM on Web Conference 2025 (WWW '25)*. 1536–1547. doi:10.1145/3696410.3714725
- [38] Xian Zhan, Tianming Liu, Lingling Fan, Li Li, Sen Chen, Xiapu Luo, and Yang Liu. 2022. Research on Third-Party Libraries in Android Apps: A Taxonomy and Systematic Literature Review. *IEEE Transactions on Software Engineering* 48, 10 (2022), 4181–4213. doi:10.1109/TSE.2021.3114381
- [39] Xiangyu Zhang, Lingling Fan, Sen Chen, Yucheng Su, and Boyuan Li. 2023. Scene-Driven Exploration and GUI Modeling for Android Apps. In *2023 38th IEEE/ACM International Conference on Automated Software Engineering (ASE)*. 1251–1262. doi:10.1109/ASE56229.2023.00179

Received 2026-01-30; accepted 2026-04-16