User Interaction Data in Apps: Comparing Policy Claims to Implementations

Feiyang Tang[0000−0002−8720−6743] and Bjarte M. Østvold[0000−0001−6922−4027]

Norwegian Computing Center
N-0314 Oslo, Norway
{feiyang,bjarte}@nr.no

Abstract. As mobile app usage continues to rise, so does the generation of extensive user interaction data, which includes actions such as swiping, zooming, or time spent on a screen. Despite the significant insights that can be learned from such data, it is often collected by apps and their services without sufficient disclosure in their privacy policies. A common issue is that many apps classify this data as non-personal, a stance that is controversial given its potential to reveal personal details when aggregated.

In response to this issue, we propose an automated approach to check a privacy policy in the following respect: We compare the policy's claims about the app's collection of user interaction data to the actually implemented collection through static analysis of the app. This process allows us to identify inconsistencies in the claims and also to study general collection practices for user interaction data across apps. Via an improved comparison between data collection claims and actual implementations, our approach aims to enhance transparency, foster trust between app developers and users, and contribute to a more informed discussion on the classification of user interaction data.

Keywords: Mobile Apps · Transparency · Trust · Interaction Data · Privacy Policy

1 Introduction

The growing use of mobile technology has resulted in an increased reliance on mobile applications, which are now a fundamental part of our daily lives. While these apps offer many conveniences, they also bring up privacy concerns, especially when it comes to the collection and use of user interaction data.

User interaction data, which includes actions like swiping, zooming, or clicking, may not directly identify an individual. Still, once aggregated, it can reveal important information about users' behaviors and preferences and further enable user profiling. While this data assist app developers in enhancing their services and customizing their offerings, the extensive nature of its collection raises issues regarding privacy, transparency, and ethics. An emerging concern is the ambiguity in the privacy policies of mobile apps, which, if left unaddressed, could undermine user trust and discourage app usage.

Our research aims to tackle these challenges by proposing an automated approach that fact-checks privacy policy statements with the results of static analysis from the app's code. This approach involves a systematic comparison of the actual data collection practices encoded in the app's code with the practices outlined in the privacy policy. By doing so, we strive to elevate transparency and equip users with the necessary knowledge to make informed decisions regarding app usage.

Our aim is to enhance transparency in data collection practices, establish trust between app users and organizations, and potentially influence regulatory bodies to set clearer guidelines for user interaction data collection.

This paper aims to answer the following research questions:

1.1 Research Questions

1. What claims do app privacy policies make concerning the collection of user interaction data?
2. What insights can be derived from analyzing app implementations in light of policy claims?
3. How can we automate the examination of the transparency of collection claims in privacy policies based on the evidence obtained through static analysis?

1.2 Contributions

Taking the above research questions into account, our research makes several contributions:

1. We introduce an automated claim extractor and classifier for processing privacy policies. Using natural language processing methods supplemented by targeted keyword searches, this technique extracts and categorizes claims about user interaction data collection.
2. We construct a static analyzer and an evidence classifier. They extract and categorize user interaction data collection details directly from app implementations.
3. We compare the labeled collection claims, extracted from privacy policies, with the labeled collection evidence derived from the application's code. This comparison provides a deeper understanding of the transparency of data collection practices.
4. Building upon these components, we conduct a study of 100 popular mobile apps. Our objective is to analyze and identify patterns in user interaction data collection, contributing to a broader understanding of this practice.

Our two-fold approach, encompassing privacy policy analysis and application code analysis, is depicted in Fig. 1.

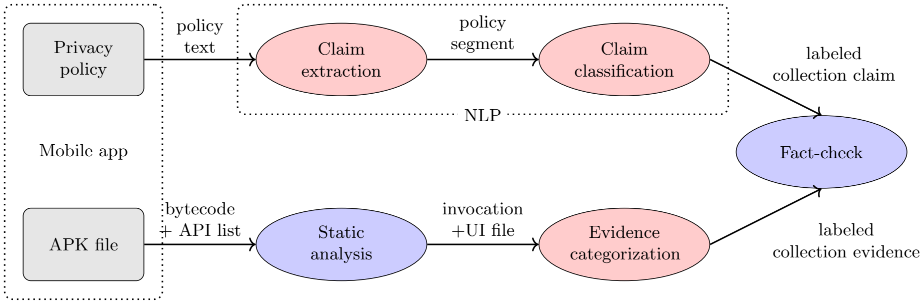

Fig. 1: Overview of the Approach

2 Motivation

We constantly notice inconsistencies between what the privacy policies claim and the actual data collection practices [4], raising questions about transparency.

One of the significant concerns involves the labeling of user interaction data. Many apps categorize this as “non-personal data” in their privacy policies [7,6]. However, recent studies demonstrate the potential to identify individual users from aggregated behavioral data, challenging its non-personal” classification. This calls for more stringent and accurate data categorization protocols.

Furthermore, as regulators emphasize data privacy [10], the demand for reliable evidence of data collection practices becomes crucial. The importance of transparent data practices extends to various sectors, including agriculture and healthcare, indicating the universality of this issue [14,11,5].

Therefore, in response to these challenges, we aim to augment transparency in data collection practices, re-evaluate the concept of personal data, and propose an automated method to cross-verify privacy policy claims with actual app behaviors [1]. Our motivation lies in the need for an enhanced understanding of what is truly personal data and to provide an automated approach to aid in this process.

3 Analyzing Collection Claims from Privacy Policies

In an attempt to assess the transparency of data collection practices stated in privacy policies, we developed a two-tiered methodology. This strategy is specially designed to extract and classify claims related to user interaction data collection, a facet less explored in privacy policy analysis.

This methodology aims to answer three primary questions:

- *Does the privacy policy mention user interaction data collection?*
- *If so, what types of user interaction data are claimed to be collected?*
- *What techniques are claimed to be used for this data collection?*

3.1 Claim Extraction

The first phase of our approach identifies whether user interaction data collection is mentioned within a privacy policy. Instead of conventional keyword-based approaches, this extractor utilizes semantic context to accommodate the diverse ways such claims can be articulated.

The APP-350 Corpus was utilized in this stage [17]. This corpus comprises 350 Android app privacy policies annotated for privacy practices. However, the existing annotations primarily focused on personal data collection, which didn't coincide with our emphasis on user interaction data collection. Therefore, we conducted our own manual annotations.

Key Findings Our review of the 350 app privacy policies yielded several key findings that offered insights into the disclosure practices regarding user interaction data collection.

By doing manual annotation we found that out of the 350 analyzed apps, 294 mentioned the collection of user interaction data at varying detail levels. However, of these 294, only 57% (169/294) of the policies provided more specifics than a mere mention of "data" or "information." Upon segmenting the privacy policies into sentences, we annotated 3,661 sentences as relevant to user interaction data collection from a total of 42,797 sentences. Table 1 presents the most frequently occurring bigrams within these annotated sentences.

Table 1: Top 10 Most Frequent Bigrams

Bigram	**Frequency**
Your Information	3,295
Our Service	2,941
Your Data	2,892
Third Party	1,788
Help You	1,214
Improve Service	947
Automatic Collection	422
Tracking Technology	402
Interact With	346
Collect Information	281

Transparency about user interaction data collection varied significantly across apps. Although 294 policies referenced such data collection, the details were often obscured by general phrases like "*we collect data to improve our service.*" Our bigram analysis highlighted the common use of third-party services in the data collection process. These services, often referred to as "*tracking technology*", are employed to automatically collect data purportedly to enhance services. "*Google Analytics*", a prominent third-party analytics service, was frequently observed in our bigram analysis, underscoring its vital role in user interaction data collection.

3.2 Claim Classification

The second phase of our approach classifies the claims. This model is innovative in its ability to categorize claims according to user interaction data types and collection techniques. Unlike traditional binary classifiers, it acknowledges that a single sentence may convey multiple types of data and collection techniques.

Key Findings In our examination of the 169 privacy policies that offered more explicit information about user interaction data collection, we found that only 56 policies clearly stated the collection techniques, such as *"the times you click a page"* or *"the time you spend watching content"*.

To standardize vocabularies and taxonomy for classification purposes, we utilized data types and collection techniques from our previous work, known as collection vocabularies [12]. These vocabularies included six types of interaction data and an additional category named device data, which we observed is commonly collected alongside interaction data. The frequencies with which of different data collection types are mentioned in the policies are shown in Table 2.

The descriptions given by the apps about their collection techniques were often vague. Of the 56 apps that vaguely mentioned the techniques used, all referred to frequency, representing 100% of this subgroup. A substantial but smaller portion, 48% (27 out of 56), mentioned duration, using phrases like *"time spent watching"* or *"length of service use"*. However, only a mere 1.8% (1 out of 56) of these apps mentioned motion.

Transparency was lacking in the descriptions of user interaction data collection types and techniques. Of the 294 apps that acknowledged data collection, a majority, 84% (248 out of 294), categorized the collected data as *"non-personal data"*, without providing further details. Such categorization seemed to be used to justify sensitive actions like *"aggregation"*, a method mentioned by 43% (126 out of 294) of these apps, and *"transfer to third-party services"*, an action mentioned by 68% (199 out of 294). Furthermore, almost half, 48% (141 out of 294), acknowledged using *"automatic collection"* methods.

4 Analyzing Collection Evidence from Application Code

Once the collection claims from the privacy policy have been extracted, we seek to validate these claims by investigating the application code for tangible signs of data collection. Our attention is primarily devoted to identifying and categorizing the embedded data collection techniques within the mobile application. The approach we adopt for static analysis explicitly targets user interface (UI) elements and the invocations to analytics libraries from these UI elements. Following identification, these elements are classified based on a predefined

Table 2: Frequency of different types of data collection (out of 169 apps)

Data Type	**Frequency**
App Presentation	98%
Categorical	60%
User Input	45%
Binary	17%
Gesture/Composed Gesture	2%
Device Data [1]	92%

collection vocabulary that we have introduced in [12]. This vocabulary was generated through a meticulous examination of all Android UI widgets and it captures a broad range of user interaction data types and collection techniques.

The collection vocabulary not only allows for a structured classification of data collection instances but also facilitates the mapping between the collection evidence found in the code and the claims made in the privacy policy. The usage of this comprehensive vocabulary ensures that we can conduct a granular comparison later in the fact-checking process.

Through our analysis, we aim to answer the following questions:

- *Which analytics libraries are being utilized by the mobile app?*
- *What types of user interaction data are being collected?*
- *Which techniques are employed for data collection?*

4.1 Analytics Library Identification

In the first stage of our code analysis, we focus on identifying the analytics libraries that are used by the mobile application. It is common for applications to utilize such libraries to gather and analyze user interaction data, providing developers with valuable insights into user behavior.

To achieve this, we target a set of popular analytics libraries as our initial point of focus. These libraries are often integral to tracking user interactions and facilitating data collection. Hence, recognizing these libraries' invocations serves as an efficient guide to pinpoint locations where user interaction data collection is likely to take place.

Our analysis primarily focuses on the classes that engage with UI elements, carefully examining the imported analytics libraries along with their respective method invocations. We constrain our investigation to a selected set of methods belonging to popular analytics libraries that are frequently utilized for data collection. In this context, we adopted the list of the top 20 analytics services for Android applications listed on AppBrain[2]. Prior understanding of these frequently used analytics libraries and their APIs forms a crucial foundation for this stage of our analysis.

4.2 Categorizing Data Types and Collection Techniques

Following the identification of analytics libraries, our objective is to establish links between the UI elements and the corresponding bytecode that manages user interactions. UI actions such as button presses trigger specific methods within the bytecode. Thus, we delve into both XML files, which define the UI elements, and the bytecode, which dictates the actions corresponding to these elements. The examination of these components often provides insights into the type of user interaction data being collected.

[2] https://www.appbrain.com/stats/libraries/tag/analytics/android-analytics-libraries

Table 3: Types of user interaction data and corresponding main Android UI elements

Interaction Data Types	**Android UI Elements**
App Presentation	View (TextView, VideoView, WebView, etc.)
Binary	Button (ImageButton, CheckBox, etc.)
Categorical	AbsSpinner (Spinner), CompoundButton (RadioButton, Switch), RatingBar
User Input	TextView (EditText, AutoCompleteTextView, SearchView)
Gesture	GestureDetector, ViewPager, SwipeRefreshLayout
Composite Gestures	GestureDetector (ScaleGestureDetector)

For instance, consider a simple scenario where a Firebase Analytics library is employed in an Android app. A button click in the UI represented as `<Button android:onClick="buttonClick"/>` in the XML file, would trigger a corresponding `buttonClick(View view)` method in the Java code. The interaction with the analytics library within this method could look something like this:

```
public void buttonClick(View view) {
    FirebaseAnalytics mFAnalytics = FirebaseAnalytics.getInstance(this);
    Bundle params = new Bundle();
    params.putString("Button_name", "button1");
    params.putString("Action", "click");
    mFAnalytics.logEvent("ButtonClick", params);
}
```

Here, an invocation to the Firebase Analytics library occurs whenever the button is clicked, recording the button's name and the associated action. This example highlights that click data is collected each time the button is clicked.

Though this method generally proves effective in discerning the types of user interaction data being collected, it is important to note that some complexities in the bytecode may obscure certain data collection events. Additionally, data collected outside of standard UI interactions, such as device-generated data or data from non-UI sources, may not be captured by this approach. Building upon the successful linking of UI elements to their corresponding analytics library invocations, we categorize the extracted data based on predefined interaction data types and collection techniques. Our initial focus is on the types of user interaction data, where we aim to classify the data according to their corresponding UI elements. Table 3 presents a classification of interaction data types associated with common Android UI elements.

In the table, the main Android UI elements represent the core classes or interfaces in the Android UI hierarchy. For instance, View is a fundamental class for UI widgets in Android, and the various UI elements like TextView, VideoView, and WebView are its subclasses, hence included as its subcategories.

In this process, we perform an inspection of each UI element across the XML files, which define the UI, and the code files that handle these UI elements. Accordingly, the type of user interaction data is ascertained based on the functionality attributed to the UI elements.

Identification of Collection Techniques Our approach to identifying the collection techniques for user interaction data consists of two components: rules-based identification using predefined criteria, and criteria obtained from a detailed analysis of popular analytics libraries' documentation.

In rules-based identification, we create a set of heuristics centered on invocations of Android or Java methods, which are associated with different collection techniques. For instance, the "frequency" technique can be inferred from the event logging invocation. Techniques like "duration" collection can be suggested by invocations of methods from the Java `Timer` class or `android.os.SystemClock.elapsedRealtime()`. Similarly, "motion details" collection can be inferred from methods from the `MotionEvent` class, such as `getPressure()`, `getX()`, and `getY()`.

The second component of our approach involves using criteria obtained from the documentation of widely-used analytics libraries, such as Firebase Analytics and Mixpanel. Once the specific API methods used for different collection techniques in these libraries are identified, they are added to our categorization list. For instance, Firebase Analytics' `logEvent()` method, with parameters like `select_content` and `view_item`, can log the frequency of user interactions. On the other hand, Mixpanel uses the `track()` method with event names to record frequency. For recording duration, Firebase Analytics uses the `user_engagement` event, capturing user engagement duration, while Mixpanel provides the `time_event()` method to time events' duration.

While this approach provides a systematic and informed means to identify collection techniques, it also has limitations. For example, if an app uses a custom package without Java or Android method invocations, or if it uses a third-party service not included in our list, our categorization method may not accurately identify the collection technique used.

5 Fact-Checking Privacy Policy Claims

Upon completing the static analysis and organizing the privacy policy collection claims, we have the necessary foundation to perform a fact-checking analysis on these claims. The goal of this process is to detect any inconsistencies between the data collection practices described in the policy and the actual practices observed in the application code. The process unfolds in two stages:

5.1 Mapping Interaction Data Types and Collection Techniques

In the first stage, we create a mapping between the types of data outlined in the privacy policy and the equivalent interaction data types pinpointed during our static analysis. A similar mapping is constructed for each collection technique stated in the policy and the corresponding technique identified within the application code.

For instance, suppose a privacy policy declares, "*We collect the content you provide*", implying the collection of user-input data. During our satic analysis, we identify the invocation of `EditText` elements in the application code, which signifies user input in Android. We then form a mapping between the phrase "*We collect the content you provide*" from the privacy policy and the `EditText` elements found in the code.

In another case, if the policy statement indicates, "*We track how long you spend on our services*", which suggests the usage of a duration-based collection technique. Suppose we identify the invocation of `android.os.SystemClock.elapsedRealtime()` in the code, which measures elapsed time, a mapping is established between the policy phrase "*We track how long you spend on our services*", and the `android.os.SystemClock.elapsedRealtime()` invocation in the code.

These mappings provide a basis for comparing the privacy policy's claims to the actual evidence in the code, allowing us to assess the consistency between policy declarations and the application code's actual practices.

5.2 Interaction Consistency Analysis

Having established the mappings, we can compare the data types and collection techniques from the privacy policy to those discovered in the code. This allows us to calculate the **Interaction Consistency Rate**, which measures the extent of consistency between the collection evidence identified in the static analysis (categorized by data type and collection technique) and the corresponding claims in the privacy policy. This rate represents the proportion of collection evidence found in the code that is accurately claimed in the policy.

An inconsistency may arise if, for example, our static analysis uncovers `EditText` invocations, but there is no mention of "user input data" in the app's privacy policy. It should be noted that our analysis focuses on correlating claims made in the privacy policies with evidence gleaned from our static analysis. This means that if data collection is linked with a UI element that falls outside the scope of our static analysis, such collection will not be included in our investigation.

5.3 Context Consistency Analysis

The second stage of our analysis involves a context-based examination to comprehend the circumstances under which user interaction data is collected. Our motivation for conducting a context-based analysis is based on our preliminary observation from the APP-350 dataset, where 74% of the policy sentences related to user interaction data collection also described the context, for example, "We collect information on how you interact with our service *when you are making a purchase.*"

To accomplish this, we review the application's code and identify unique contexts under which data collection takes place. The contexts we consider here are confined to those directly linked with identifiable criteria in the bytecode, thereby limiting our scope to certain discernible contexts.

Table 4: Catalogue of Contexts for User Interaction Data Collection

Context	**Identifiable Criteria in Code**
Viewing Content	Invocation of certain View UI elements (e.g., TextView, ImageView, WebView).
Making Purchase	Calls to Android Google Play payment service APIs.
Location-Based Services	Invocation of Android Location APIs.
Interacting with Media	Calls to media-related APIs (e.g., Media Player, Media Recorder).
Search	Invocation of SearchView UI elements.
Notifications	Interactions with NotificationManager API.
Accessing User Profile	Invocation of User Profile related APIs (e.g., AccountManager).
Sensor-based Features	Use of Android Sensor APIs.
Communication Features	Use of communication-related APIs (e.g., TelephonyManager, SmsManager).
Gameplay Interactions	Calls to APIs related to gameplay, typically seen in game apps.
Customization Features	Invocation of APIs related to customization features (e.g., changing app theme).

Our approach to construct a context catalogue was developed through a thorough examination of select Android applications. We initially selected a representative sample of 25 applications from five categories within the Google Play Store in Germany [3]. These applications underwent a static analysis where we scanned the bytecode for instances of user interaction data collection, noting the specific contexts under which they occurred.

Through this process, we identified and organized recurring contexts across the different applications. These common contexts, indicative of typical scenarios associated with user interaction data collection are developed into a generalized catalogue. While not comprehensive, this catalogue, as presented in Table 4, provides an informative overview of the most common user actions and application states where interaction data collection is likely to occur.

Based on this catalog, we calculate the **Context Consistency Rate**, which measures the degree of consistency between the data collection contexts identified in the static analysis and those outlined in the privacy policy. This rate indicates the proportion of collection contexts found in the code that are accurately represented in the policy.

We recognize that our catalogue cannot encapsulate all possible contexts due to the complexity and diversity of user interactions and app functionalities. Furthermore, our policy claim checks rely on language model-assisted vocabulary matching, which might not guarantee absolute precision. These factors should be considered when interpreting the Context Consistency Rate.

[3] The German store was chosen for its variety and adherence to the General Data Protection Regulation, ensuring well-constructed privacy policies. https://play.google.com/store/apps?hl=en_US&gl=DE

6 Experiment

In this section, we present the results of a large-scale analysis conducted on a set of 100 Android applications. Through this comprehensive examination, we aim to gain insights into the landscape of user interaction data collection practices as reflected in their privacy policies and underlying code. This analysis forms the basis of our discussion on the consistencies and discrepancies between policy claims and actual code execution.

6.1 Setup

Our experimental analysis is based on a set of Android applications obtained from the Google Play Store in Germany. We selected the top 100 applications from 10 distinct popular categories, ensuring no overlaps among the categories and the chosen applications. We used two key selection criteria: (1) the categories and apps should be disjoint to avoid overlap and redundancy, and (2) every app must have a corresponding English privacy policy webpage linked in its "Data Safety" section. This selection process ensured a diverse representation of applications and categories, each adhering to the General Data Protection Regulation.

In this experiment, we used two primary metrics to assess the consistency of the data collected in the privacy policies with our static analysis results: the Interaction Consistency Rate and the Context Consistency Rate.

The Interaction Consistency Rate measures the degree of alignment between the types of interaction data identified in our static analysis and those stated in the privacy policies. Similarly, the Context Consistency Rate quantifies the level of agreement between the contexts in which data collection occurs, as identified in the static analysis, and the scenarios described in the privacy policies.

Additionally, we introduce two coverage rates to capture the completeness of our static analysis method. The Interaction Consistency Coverage Rate and the Context Consistency Coverage Rate indicate the proportion of privacy policy claims about interaction data types and collection contexts that our static analysis method was unable to detect. These coverage rates, thus, serve as indicators of the potential limitations of our static analysis approach, highlighting areas that could be missed or overlooked in the code.

6.2 Overview of User Interaction Data Collection Practices

Firstly, we provide an overview of the user interaction data collection practices as stated in the privacy policies and as evidenced in the application code of the 100 analyzed apps.

As demonstrated in Fig. 2, we found that 14 out of the 100 apps do not mention any form of user interaction data collection in their privacy policies. Roughly a third (29) of the apps acknowledge the existence of data collection but fail to provide specifics regarding the type of data collected or the method of collection. For instance, these policies may include vague statements such as, "*we* *use statistical tools to collect non-personal data such as usage details.*" Notably, the policies with more detailed descriptions of data collection practices tend to focus on the types of data collected rather than the collection techniques.

Fig. 3 and Fig. 4 showcase the distribution of interaction data types and collection techniques identified in our static analysis of the apps' implementation. It is evident that app presentation data and binary data, such as screen content and button clicks, are the most frequently collected types of interaction data. It is also noteworthy that user input data is commonly collected, often within the context of user preferences and surveys.

In terms of data collection techniques, apart from frequency, duration also emerges as a prevalent method. This observation is significant considering that many apps refrain from mentioning duration-based data collection in their privacy policies, thereby highlighting a transparency issue.

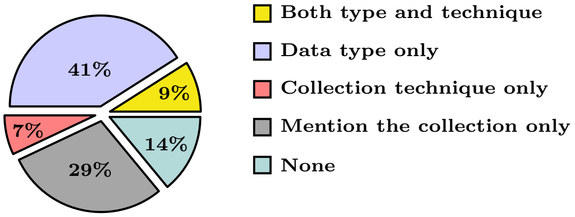

Fig. 2: Policy claims completeness regard to interaction data collection

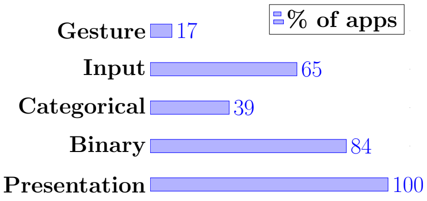

Fig. 3: Data type distribution

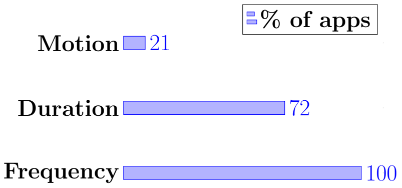

Fig. 4: Collection technique distribution

Our analysis identified five categories of apps that most frequently engage in user interaction data collection: *social, entertainment, shopping, gaming*, and *lifestyle*. The extent of data collection in these categories can be attributed to two key factors. First, the intrinsic characteristics of the category, such as social networking and entertainment, necessitate understanding user behavior for the personalization of services. Second, the complexity of functionality in certain categories, like gaming, often requires learning from user interactions to optimize user experiences. Likewise, lifestyle apps might need to track user actions within the app to function effectively.

It is worth noting that almost all apps, across categories, engage in some form of user interaction data collection. However, the level of transparency in detailing such practices within their privacy policies varies widely. The majority of these policies lack completeness, indicating a trend of incomplete disclosure about user interaction data collection practices. This highlights the urgent need for more transparent and detailed communication about these practices in app privacy policies.

6.3 Case Study: In-depth Analysis of Four Popular Apps

Following the general overview, we delve into a more detailed exploration by conducting a case study on four popular apps from the German Google Play Store: WetterOnline (a local weather app), Temu (a Chinese e-commerce platform), Poe (a chatbot app developed by Quora), and Plant App (a plant identification app). Except for Temu, all of these apps provide specific services that seemingly should not involve extensive user interaction data collection. The purpose of this case study is to demonstrate how apps, which may be considered benign, can still have non-transparent policy claims and engage in substantial user interaction data collection. Table 5 provides an examination of their policy claims alongside our fact-checking results based on the static analysis.

One striking observation is the lack of specificity in the contexts of data collection described in the policies of these apps. Many use a vague term such as "when you interact with our service". Interestingly, although one might suspect Temu, an e-commerce app, to collect a significant amount of user interaction data, our analysis confirms this suspicion but also reveals that Temu's privacy policy is relatively transparent about their data collection practices.

In contrast, the simpler apps WetterOnline and Plant App provide only limited information regarding their data collection types and techniques. This limitation is even more pronounced in the case of the Poe chatbot, which offers almost negligible information related to user interaction data collection. These examples underscore the importance and need for more transparent claims regarding user interaction data collection in privacy policies.

Furthermore, we noted that many privacy policies label their user interaction data collection as automatic and accumulative, associating it with an anonymized identifier and stating it is used exclusively for commercial purposes. They often categorize this data as non-personal. However, the vast amount of automatically collected behavioral data, when combined with the collected event-specific values and a unique machine-generated identifier, raises questions about whether such combined data can indeed be classified as non-personal. This issue underlines the need for a more nuanced understanding and categorization of user interaction data in privacy policies.

7 Related Work

The analysis and improvement of privacy policies in mobile applications have been the focus of numerous studies. Various NLP techniques have been explored to automatically process and understand the texts of privacy policies and to help

Table 5: Result of fact-checking data collection claims w.r.t. evidence for the popular 4 apps in Google Play. The red text indicates types of user interaction data missing from the privacy policy/collection claims, while the blue text indicates undisclosed techniques of collection.

App	Policy Claims	Collection Evidence
Wetter Online	The goal of usage measurement is to determine the intensity of use, the number of uses and users of our application, and their surfing behavior statistically. The information about the use (..., the site visited, date and time of your visit. The event-driven data collection ... is triggered by activities such as installation and start of the app, ..., and in-app purchases as well as the receipt, the swipe and the opening of push-messages and the opening and updating of the app by means of a dynamic link. For each of these events the number of visits, the number of users triggering the event and, if available, the value of the events is collected.	Interaction Consistency Rate: data type 3/4; collection technique 1/2. Context Consistency Rate: 1/6. Data types: presentation, categorical, binary, user input. Collection techniques: frequency, duration. Context: viewing content, location, search, notification, sensor-based, customization.
Temu	Online activity data, such as pages or screens you viewed, how long you spent on a page or screen, the website you visited before browsing to the Service, navigation paths between pages or screens, information about your activity on a page or screen, access times and duration of access, and whether you have opened our emails or clicked links within them.	Interaction Consistency Rate: data type 3/5; collection technique 3/3. Context Consistency Rate: 1/10. Data types: presentation, categorical, binary, user input, gesture. Collection techniques: frequency, duration, motion. Context: viewing content, purchase, location, media, search, notification, user profile, sensor-based, communication, customization.
Poe	Our third party LLM providers and third party bot developers may receive details about your interactions with Poe (including the contents of your chats, upvotes, etc.) to provide and generally improve their services, which they may process in their legitimate business interests.	Interaction Consistency Rate: data type 1/3; collection technique 0/2. Context Consistency Rate: 1/6. Data types: presentation, binary, user input. Collection techniques: frequency, duration. Context: viewing content, location, search, notification, communication, customization.
PlantApp	During your visits, we may use software tools such as JavaScript to measure and collect session information including page response times, download errors, length of visits to certain pages, page interaction information (such as scrolling, clicks, and mouse-overs), and methods used to browse away from the page.	Interaction Consistency Rate: data type 2/5; collection technique 3/3. Context Consistency Rate: 1/8. Data types: presentation, categorical, binary, user input, gesture. Collection techniques: frequency, duration, motion. Context: viewing content, purchase, location, media, search, notification, sensor-based, customization.

users understand these policies more effectively [13,9]. This body of research is similar to ours in its overarching objective to clarify privacy policies, but our study distinguishes itself by specifically addressing the collection of user interaction data.

Tools such as PrivacyFlash Pro [16] and AutoCog [8] have been proposed in prior studies to check the compliance of privacy policy disclosures with actual app behavior. However, these tools have primarily targeted personal data, leaving the collection practices around user interaction data largely unaddressed. Our work complements these earlier efforts by focusing on this less-explored area of data collection.

Static analysis techniques have been broadly applied in the domain of mobile app security and privacy [2,3,15]. While these studies have made important strides in analyzing app bytecode, identifying data leaks, and detecting privacy violations, they have generally not zeroed in on the collection of user interaction data—a dimension of data collection that is often overlooked in privacy policies and analyses.

Our research extends the scope of these prior efforts by providing an automated method for comparing privacy policy disclosures with the actual behaviors of apps, specifically in relation to user interaction data collection. Our goal is to enhance transparency and trust within the mobile app ecosystem. Our work, while building on prior research, contributes to the existing body of knowledge by filling in this identified gap.

8 Conclusion and Future Work

Through this research, we have examined the practices of user interaction data collection in mobile applications with a focus on transparency. Our automated approach has enabled a direct comparison between privacy policy claims and the actual implemented data collection activities, as identified through static analysis. The findings underscore the need for enhanced transparency and better policy communication in the realm of mobile applications.

However, the limited scope and the claim-to-evidence mapping based on a self-constructed list of Android UI types present inherent limitations. Future research should expand and refine these methods, ensuring wider coverage of applications, platforms, and analytics services for a comprehensive understanding of data collection practices.

The common classification of user interaction data as non-personal needs further discussion due to its potential to profile a person when aggregated. Advancing user awareness and fostering responsible practices among developers are essential for a transparent mobile app industry. Our study calls for these actions, contributing to the discourse on responsible and transparent user interaction data collection practices.

Acknowledgement

This paper is an extended version of work published in [12]. This work is part of the Privacy Matters (PriMa) project. The PriMa project has received funding from European Union's Horizon 2020 research and innovation program under the Marie Skłodowska-Curie grant agreement No. 860315.

References

1. Acquisti, A., Brandimarte, L., Loewenstein, G.: Privacy and human behavior in the age of information. Science **347**(6221), 509–514 (2015)
2. Avdiienko, V., Kuznetsov, K., Gorla, A., Zeller, A., Arzt, S., Rasthofer, S., Bodden, E.: Mining apps for abnormal usage of sensitive data. In: The 37th IEEE international conference on software engineering. vol. 1, pp. 426–436. IEEE (2015)
3. Enck, W., Gilbert, P., Han, S., Tendulkar, V., Chun, B.G., Cox, L.P., Jung, J., McDaniel, P., Sheth, A.N.: Taintdroid: an information-flow tracking system for realtime privacy monitoring on smartphones. ACM Transactions on Computer Systems (TOCS) **32**(2), 1–29 (2014)
4. Fair, R., et al.: Privacy and data protection by design - from policy to engineering. arXiv preprint arXiv:1901.07159 (2019)
5. Kosinski, M., Matz, S.C., Gosling, S.D., Popov, V., Stillwell, D.: Facebook as a research tool for the social sciences: Opportunities, challenges, ethical considerations, and practical guidelines. American psychologist **70**(6), 543 (2015)
6. de Montjoye, Y.A., Hidalgo, C.A., Verleysen, M., Blondel, V.D.: Unique in the crowd: The privacy bounds of human mobility. Scientific reports **3**, 1376 (2013)
7. de Montjoye, Y.A., Radaelli, L., Singh, V.K., Pentland, A.S.: Unique in the shopping mall: On the reidentifiability of credit card metadata. Science **347**(6221), 536–539 (2015)
8. Qu, Z., Rastogi, V., Zhang, X., Chen, Y., Zhu, T., Chen, Z.: Autocog: Measuring the description-to-permission fidelity in android applications. In: Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security. pp. 1354–1365 (2014)
9. Ravichander, A., Black, A.W., Norton, T., Wilson, S., Sadeh, N.: Breaking Down Walls of Text: How Can NLP Benefit Consumer Privacy? In: Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing. vol. 1 (2021)
10. Schwartz, P.M.: The EU-US privacy collision: a turn to institutions and procedures. Harv. L. Rev. **126**, 1966 (2012)
11. Tahir, M., Sardaraz, M., Mehmood, Z., Muhammad, S.: Cryptoga: a cryptosystem based on genetic algorithm for cloud data security. Cluster Computing **24** (2021)
12. Tang, F., Østvold, B.M.: Transparency in app analytics: Analyzing the collection of user interaction data. arXiv preprint arXiv:2306.11447 (2023)
13. Tesfay, W.B., Hofmann, P., Nakamura, T., Kiyomoto, S., Serna, J.: PrivacyGuide: Towards an Implementation of the EU GDPR on Internet Privacy Policy Evaluation. In: Proceedings of the Fourth ACM International Workshop on Security and Privacy Analytics. p. 15–21. IWSPA '18 (2018)
14. Wiseman, L., Sanderson, J., Zhang, A., Jakku, E.: Farmers and their data: An examination of farmers' reluctance to share their data through the lens of the laws impacting smart farming. NJAS-Wageningen Journal of Life Sciences **90** (2019)
15. Zhang, X., Wang, X., Slavin, R., Breaux, T., Niu, J.: How does misconfiguration of analytic services compromise mobile privacy? In: Proceedings of the ACM/IEEE 42nd International Conference on Software Engineering. pp. 1572–1583 (2020)
16. Zimmeck, S., Goldstein, R., Baraka, D.: PrivacyFlash Pro: Automating Privacy Policy Generation for Mobile Apps. In: NDSS (2021)
17. Zimmeck, S., Story, P., Smullen, D., Ravichander, A., Wang, Z., Reidenberg, J.R., Russell, N.C., Sadeh, N.: Maps: Scaling privacy compliance analysis to a million apps. Proc. Priv. Enhancing Tech. **2019**, 66 (2019)