Hindawi
Mobile Information Systems
Volume 2021, Article ID 3452700, 19 pages
https://doi.org/10.1155/2021/3452700

Review Article

Influencing User's Behavior Concerning Android Privacy Policy: An Overview

Ming Di,[1,2] **Shah Nazir**,[3] **and Fucheng Deng**[2]

[1]*School of Journalism and Communication, Wuhan University, Wuhan 430072, China*
[2]*School of Information Management, Wuhan University, Wuhan 430072, China*
[3]*Department of Computer Science, University of Swabi, Swabi, K.P., Pakistan*

Correspondence should be addressed to Ming Di; mingdi7@whu.edu.cn and Shah Nazir; shahnazir@uoswabi.edu.pk

Received 14 October 2021; Accepted 17 November 2021; Published 29 November 2021

Academic Editor: Sikandar Ali

The wide-ranging implementation of Android applications used in various devices, from smartphones to intelligent television, has made it thought-provoking for developers. The permission granting mechanism is one of the defects imposed by the developers. Such assessing of defects does not allow the user to comprehend the implication of privacy for granting permission. Mobile applications are speedily easily reachable to typical users of mobile. Despite possible applications for improving the affordability, availability, and effectiveness of delivering various services, it handles sensitive data and information. Such data and information carry considerable security and privacy risks. Users are usually unaware of how the data can be managed and used. Reusable resources are available in the form of third-party libraries, which are broadly active in android apps. It provides a diversity of functions that deliver privacy and security concerns. Host applications and third-party libraries are run in the same process and share similar permissions. The current study has presented an overview of the existing approaches, methods, and tools used for influencing user behavior concerning android privacy policy. Various prominent libraries were searched, and their search results were analyzed briefly. The search results were presented in diverse perspectives for showing the details of the work done in the area. This will help researchers to offer new solutions in the area of the research.

1. Introduction

The wide-ranging implementation of Android applications used in various devices from smartphones to intelligent television, has made it thought-provoking for developers. The permission granting mechanism is one of the defects imposed by the developers. Users of mobile are not aware of their proper use. Security and privacy are the primary concerns of mobile applications. Research work has focused on different aspects of mobile applications and devised solutions. In the context of Android applications, a novel process for evaluating privateer's protection approaches is proposed [1]. The device, which has recently been tested to include worrying examples of usage, dramatically simplifies the path toward comprehending the security repercussions of inserting in apps. The appliance is made to grow capability. As a result, inclines in the approach can be easily integrated to improve the consistency of quality and sufficiency. To examine the relative usefulness of the two permissions interfaces was presented in the study [2]. People were recruited through Amazon Mechanical Turk for online research. Users do not read permissions for three reasons: they are unconcerned about or unaware of permits; instead, they rely on other measures to choose whether or not to download an app. They have faith in the programmer and system based on variables such as the number of downloads and user ratings, and they have confidence in the developer and platform to screen out highly unsafe programs. The Android 6 UI, according to participants, is more visually attractive than Android5. A proactive user-oriented approach is recommended to enhance user awareness of the privacy risks connected with Android applications to grant access [3]. An effectual privacy assessment approach is described, which evaluates users' privacy connected with a set of permissions-required apps. Permissions' harshness and relative relevance, as well as their linkage, are the model's parameters. A standard severity evaluation method is used to determine severity. A data mining algorithm was used to identify association rules between permits.

The Android operating system is inspected to create Near-Field Communication (NFC) empowered applications [4]. The security of NFC is briefly discussed just as an outline of the three modes write/read, distributed, and card copying which Android's API exposes to developers. Some current Android NFC applications, like demonstrative apparatuses, contactless label control instruments, shared NFC applications, and some unusual use-cases, are also described. On mobile app privacy, thorough mapping research of Software Engineering literature is undertaken. The study aims to systematically expose the privacy practices of apps and make recommendations for how to preserve the privacy of mobile app users [5]. The objectives are to look into current software engineering application research patterns, categorize available affirmations, and make recommendations for future research. The goals further classify the state-of-the-art and list the issues that the Software Engineering research community must address. The study looked at privacy policies and the data security of depression-related mob applications [6]. The transparency of data processing practices of the mob applications acquired from the Google Play Store and iTunes was assessed and graded using the term "depression." A total of 116 eligible mobile phone applications have been discovered. The apps had only a 49% privacy policy, according to the study. Strategies ran enormously by stage, with iTunes applications being bound to incorporate one than applications from the Google Play shop.

Designers should dedicate additional time and effort to software development procedures to satisfy the expanded interest for excellent applications. With an emphasis on recent advancements, the research explores the main issues and genuine concerns in Android mobile application testing [7]. The study provides principles, rules, models, methodologies, and technology for Android application testing and outlines future perspectives. A new approach (CUPA) is offered that permits clients to oversee application admittance to Android framework assets, and private information dependent on client characterized techniques [8]. This method furnishes clients with alternatives not accessible in the Android consent framework in the establishment and the actual time of mobile application, allowing them to decrease the extent of the privacy breach. Users can regulate app behavior using the proposed approach, and application network connections lists can be included and inter-app communication. The suggested method comprises three main components that may be used to check program behavior during installation and runtime. First, a method was offered for analyzing an Android app to determine whether it is malicious or not [9]. Second, by examining permission patterns, a unique technique is suggested to discover malware-based programs. The proposed method obtains permission clusters by using the k-means algorithm to isolate malicious apps. The approach is validated by a 90 percent efficiency rate for malicious behavior. This study backs up the use of application authenticity for applications in Android phishing detection.

2. User Privacy Approach for Android Mobile Application

DroidNet uses the framework to put new apps through a trial period before granting their requests for approval. Based on peer expert user decisions, it provides suggestions about whether to approve or decline access requests to assist users in implementing limited resource access restrictions on unknown apps to protect their identity and improve resource consumption efficiency. Users can install programs in quest mode using the framework, which prompts them with resource access requests and allows them to choose whether or not to approve them [10]. A study [11] has demonstrated that giving mobile phone users more choices over their information disclosure and increasing their awareness of adverts substantially impacted privacy behaviors and views. Researchers created a privacy warning dialog that simulates actual pre-installation privacy controls screens for Android apps to carry out and manage advertisements recognition. The implications for creating privacy notification dialogs in mobile apps as well as distinct commercial methods are examined. PanGuard, an automated approach for detecting Third-party libraries (TPL) from a large number of Android APPs, is provided in this work [12]. To characterize TPLs, a novel combination of features is presented that includes packages in applications that can have both organizational and semantic data. In APPs, TPLs are isolated from the principal code, and invariants that remain unaltered during modification are identified, and the contained TPLs and versions are determined using these invariants. Extensive testing shows that PanGuard delivers significant TPL quality aspects and sustainability at the same time.

A semi-automated model is proposed to support smartphone app programmers in checking the security practices against their application code for reliability. It includes an API policy terminology map that links strategy phrases to API techniques to produce private data and dataflow analysis to diagnose disparity. According to the study, the implementation of the system relies on a series of mappings from API functionality to terms of policy and a cyber-security taxonomy [13]. Styx is a ground-breaking Android perceived communicating privacy system that provides users with information about privacy risks starting with the second security strategic viewpoint [14]. Styx's trial results are reviewed in terms of its impact on consumer perception of privacy problems and smartphone credibility, and its performance in managing risk. The study's findings imply that Styx's privacy risk information enhances the readability of security information disclosure and assists users in analyzing the privacy aspects of various applications. An AndroMalyZer framework is presented, which evaluates the application's expected privacy behavior to discover anomalous privacy behavior [15]. The framework derives semantics of desired privacy behavior from the application's description and privacy policy to identify probable privacy-related abnormalities. The trials demonstrated AndroMalyZer's feasibility and accuracy in inferring an application's predicted privacy behaviors and describing any anomalous privacy behaviors. This presentation provided a quick overview of Android permissions and their current security methods. A study [16] includes a complete assessment of the Android operating system drawbacks, their influence on end-user privacy and security, and an empirical model to resolve these concerns. Whenever it concerns program installation on Android smartphones, the results reported in the study reveal that people are unfamiliar with the most basic and essential aspects of Smartphone licensing.

User research with 26 participants was done, and it was discovered that while participants were satisfied with better privacy options, they struggled to adjust to complicated customized interfaces. In a study [17], a practical solution is proposed to assist users in sorting, suggestions, and profile management, and all require balancing absolute authority and the extra interactive complexity. The study's key conclusion is that privacy controls must allow users to control their devices and be informed about confidentiality. Research presented a privacy leak-based Android application analysis scheme with a combination attack. The linked apps are extracted after the risk components of the application are discovered. This scheme's implementation process and principles are discussed. Because it is supported by experimental evidence, the analytical strategy is viable. The difficulty and precision of application pretreatment are increased to some extent [18]. A new approach for effectively detecting and testing authority entrust flaws was constructed by integrating static code analysis, language processing, deep learning, and biological algorithm-based test creation strategies given in the study [19]. The approach finds and identifies insecure applications that reveal delighted to support other apps in an unusual way from legitimate permission entrust circumstances. To demonstrate the vulnerabilities, it also creates solid evidence intrusions and protection reports. The multi-criteria app evaluator of trust for android (MAETROID) framework is presented in this paper [20] as a way to assess the trustworthiness of Android apps in terms of secrecy and reliability. At the time of deployment, MAETROID does a multi-criteria evaluation of an app and provides a single, simple risk level rating, guiding the customer's choices as to whether or not to install a new application. A bundle of metadata acquired out from the global market that indicates the app's reliability and desirability and a collection of desired privileges are among the criteria.

A strategy was presented that uses privacy as a service level agreement rather than the transparency consciousness model. For evaluating an Ambient Assisted Living Application's proper privacy settings and managing real-time approval queries, the study [21] presented a participatory privacy protection algorithm. A case study is used to show how these algorithms function. The proposed method is also compared to the state-of-the-art Android privacy management methodology. The proposed algorithms can help users of Ambient Assisted Living Apps in digital cities safeguard their privacy and relieve them through cognitive offloading. A survey [22] was presented to address the security and privacy concerns around Android application permissions. Because a real-world Android application investigation validates the findings, the protocol has serious security consequences. However, the Android permission protocol has several flaws. The attacker can altogether bypass the permission checks in each of these scenarios. Although application permission-based malware is discussed, as the Android market grows rapidly, a new sort of malware family is forming. A complete study on quick vulnerability assessment was described [23] and thorough reverse engineering using a variety of smartphones and digital operating systems. The data leakage of the use of messaging platforms on mobile phones is examined. Android-based smartphones with the help of a particular susceptibility evaluation system set were used for this project. Evaluation and vulnerability simulations are also covered in depth to make instant messaging systems more safe and free of vulnerabilities.

3. Privacy-Preserving Categorization of Mobile Applications

A study has proposed to adopt a proactive role to users' awareness concerning misuse of personal information connected with granting rights to mobile applications [24]. A general privacy risk assessment model was proposed, which assesses the vulnerability to clients' security associated with a collection of permissions-based applications. This proactive technique was validated through an experimental analysis. The work originality stems from the fact that the security risk for a particular node held by a user changes over time based on the device's various usage, applications, and related permissions. Research has presented mobile App Reviews Summarization (MARS), which uses Google-Play-Shop user evaluations as an appropriate source for extracting and quantifying private-related allegations linked with applications [25]. MARS uses machine learning to recognize privacy-relevant reviews and classify them into a pre-defined list of confidentiality in the mobile context application. The integration of such approaches enables developers with precise details regarding privacy issues and application behavior to provide self-reports that would be impossible to identify otherwise. The students' mobile privacy behavior and attitudes are investigated and compared in the case study. In Germany and the United States, participants were observed, and an experiment was used as part of the qualitative ethnographic studies. The findings of the study show that participants' mobile privacy behavior and attitudes are nearly identical across cultures. In reality, this research identifies and distinguishes various types of mobile privacy. These classifications, which range from "mobile privacy objection" to "mobile privacy learned helplessness," demonstrate how they have a massive effect on German and American pupils' privacy behavior and attitudes [26].

This discovery has led to developing a new confidentiality classification technique for mobile applications, which is intended to explain patterns from a vast amount of usage patterns. To make app feature vectors, shuffle usage data from a large number of users to make it anonymous, then formalize data information as a dataset, identify and aggregate usage information for every application using Dynamic Time Warping, and then utilize the Dynamic Time Warping and Shape Features based on Symbolic Aggregate approXimation methods. On 3,086 apps, five machine learning techniques were employed to train and validate categorization models. SVM is the best performer, as per the results [27]. The Native Protector system, which regulates third-party in Android apps, local libraries are used is proposed [28]. The server app and the client app are the two halves of the standalone Android app. The client application holds the rest of the original app and the server application containing the libraries for offering services from the modules. Native Protector has been implemented as a prototype and successfully detects and blocks efforts by Android apps that use third-party native libraries to undertake harmful actions. In malicious assaults, Android phones can be used as a target and a tool. Individuals and corporations must be aware of the risks and take steps to avoid being exploited for nefarious ends. Mobile app developers must also pay greater attention to security issues and take responsibility for user data protection. Even if the basic security techniques recommended in this chapter do not provide total protection, they can act as a deterrent against most attempts [29]. The study has proposed a new hybrid methodology for ascertaining private information leaks that outperform existing static and dynamic approaches [30]. The notion tool implemented is Hybri-Droid, which extracts each app's models using both dynamic and static analysis methods, then refines the behavior model depending on the dynamic analysis results. According to the evaluation results, for both inter-app and intra-app interactions, Hybri-Droid is excellent at identifying security leaks. It can significantly increase data leakage detection performance when compared to previous techniques.

The research was presented with a large-scale and comprehensive examination of users' private information leakage over a nine month period which included surveillance of popular Chinese App Store apps [31]. The study's main findings are that mobile apps that access users' private information are widespread. However, the architecture of access to personal data differs slightly between various categories of applications, and applications downloading from big App stores do not always imply that the apps are safer more private. Another study was proposed which uses natural language processing techniques to translate abstract terminology in that the architecture of access to personal data differs slightly between various categories of applications. The violation detection system described by Hosseini [32] is planned to apply the automatically produced mapping. With the rising availability of private data and the widespread use of phone apps, the ability to trace relationships between app code and privacy policy requirements is becoming increasingly critical. NLP and program code analysis techniques can be used to establish automated traceability. Research has developed a model that relies on real appeals and investigates the factors that influence APP users of various experience levels' willingness to take steps to protect one's privacy and security [33]. Researchers observed disparities in the contributing differences in factors across groups of users who have and do not have expertise in APP privacy protection after conducting a questionnaire survey and doing a route analysis. This research has some implications for third party cyber security providers such as application network operators. The study proposes two hypotheses [34]: (a) a lack of understanding of the relationship between authorization requests and privacy; and (b) the installation of sensitive data-accessing programs. The examination of the acquired data, which took into account both treatments, revealed the emergence of distinctive attributes that were shared by a certain user group. This research could be seen as a first step toward more secure smartphone usage, as such features could be used to draw users' attention to privacy concerns.

Two studies were done to test the hypothesis with the aim of this research [35]. App Tracer is being used to create a dynamic analytic tool to track user interactions and sensitive resource access, as well as to conduct an online survey to see how different UI interactions alter users' assumptions about whether an app accesses sensitive resources. The findings show that user button clicks, for example, can be considered authorization, it should be possible to reduce the requirement for separate queries that are not directly related to user activities, individually, possibly when apps are first launched. The mechanism by which various apps obtain when installed on Android smartphones, access to crucial device privileges is investigated [36]. It highlighted the user's difficulties in comprehending what impact differing device restrictions have on their security. Each permission's impact is influenced by its context and use-case, and when granted numerous permissions, estimating the possible impact on client privacy becomes challenging. The research quantifies the most important human rights have an impact and takes the first steps toward a privacy impact assessment of multiple device permissions. The functionality of the program was examined in a dynamic examination, while a static analysis was carried out by looking at the code. Backdoor configuration, capture of session parameters, insecure encrypting application data transported without authentication, the usage of harmful APIs, and privileges over disclosure are all issues that need to be addressed, and unlimited connection to the databases and modules of the program were detected after executing the vulnerability analysis. The methodology presented introduces an innovative design for running a privacy test on Android-based smartphone apps [37]. The privacy framework for mobile apps is examined for flaws and a survey of proposed tools and frameworks is conducted, with an emphasis on the consequences of confidential data leaks and the cyber security threats it poses [38]. The study also offers some insight into how rogue mobile applications can exploit users' personal data for their own gain. Users are classified based on how they handle their personal information. Then we spoke about some of the problems with the permissions model framework and how Google fixed them.

The research was presented that allows Android apps to be assessed for compliance recently with the issued Google Play security standards in real-time. A novel methodology for analyzing such compliance is discussed, which relies on a successful mix of dynamic analysis and deep learning approaches. This methodology also examines if each program has private information that follows the Google Play security regulations and only accesses sensitive information once the user agrees on the policies [39]. Providing Android apps using permission control that is both versatile and perfect pleasing has been a challenging issue. A careful inspection of the Android application package file to solve such problem and the retrieved "AndroidManifest.xml" file, which lists all necessary permissions, is recommended, as well as a quick introduction to Android's authorization administrator, which demonstrates that approvals can be canceled once the apps have been installed. Yet, it does not prevent applications from exploiting resources and information [40]. The study has introduced a risk assessment approach for Android smartphone app authorization that uses dynamic analysis to determine whether or not a specific application is likely to be over-privileged [41]. The technique establishes the necessity of asking authorization indicated in the obvious of the application. The approach differs from existing approaches in that it may be used to compute the attack surface of mobile applications as well as the danger posed by over privileges. By illustrating the importance of app value in the privacy trade-off, the research verifies earlier findings and adds to our understanding of the privacy calculus' border limits [42]. The work provided is entirely experimental, and that may have implications for the internal and external validity of the conclusions. Every effort was made to make the experiments possibly realistic, and the research has strictly adhered to modification of perception criteria. According to surveys, whether downloading apps or accessing mobile websites, many people who use the Internet on their phones never check the privacy options.

4. Privacy Protection in Mobile Apps

Research works were presented to discover the discriminate and persistent elements obtained from Android APK files that are used to detect malicious apps on a large scale [43]. To accomplish this, the research extract takes a lot of features from each application and divides them into two categories. These feature sets are then used to train classifiers to detect malicious programs. Based on classifier performance, the durability of application particular and framework features are evaluated with two data sets, and the Linear Regression classifier's key characteristics are thoroughly studied. In Android, the tracking of security-related data was limited to taint tags. Taint-Droid has a simple design that allows for a maximum of thirty-two different data sources. The research introduces Kynoid, an Android-based system for monitoring and enforcing laws in real-time [44]. Kynoid deviates from the traditional taint tracking method by introducing finer granularity. It's the first significant effort to show that adaptive vulnerability management tracking is possible at this level of detail. An overview of Android and its cutting-edge security features was presented [45]. Then, based on the literature, a thorough and analytic taxonomy of Android malware hardening techniques is offered. Application developers often use hardening measures to protect against reverse engineering. The difficulties associated with them are highlighted and manifest future paths based on this in-depth survey. Then, to offer a complete picture, the trends in application hardening are shown, and a research gap summary for future studies. The knowledge of mobile app users about data gathering procedures is currently insufficient [46]. The study looks into the elements that support privacy and security in mobile applications. It adds to the knowledge about the subject. The findings demonstrate that greater degrees of consciousness and security concerns improve the motivation of smartphone applications users to participate in reducing malicious activity. According to the results, app users are inadvertently and desperately in the limelight, but this will change if they are increasingly concerned regarding their confidentiality in their ability to secure it.

Research work presented some of the security concerns addressed by the Android security architecture and reverse engineering of an Android banking app and static analysis of its code to identify flaws [47]. The work attempts to examine specific security threats in mobile networks. As a result, assaults such as malware infection and DDOS attacks are used. Finally, several recommendations are made to aid programmers in making their mobile apps more secure. A study was presented, and an X-Decaf privacy leakage detection tool and an Automatic Transparent File Encryption/Decryption (ATFed) auto-protection technique on the Android platform in mobile social networking applications (MSNAs) is recommended [48]. MSNAs from China's domestic market was examined, and it was discovered that during the development process, MSNAs failed to address the security of the user's privacy data. To keep the cache files created by MSNAs safe, an ATFed technique is provided, which allows the files to be saved in the ciphertext, and during the MSNA's operation, secured cache files are produced. During the course of the literature review, researchers assembled different research papers and conducted a comprehensive literature evaluation, resulting in a substantial body of work. The goal of the research was to present a coherent picture of current modern work that statically analyzes Android applications, which may draw conclusions about static analysis trends [49]. The review is divided into five sections: challenges addressed by methodology, basic approaches used by the authors, dynamic testing sensitivity properly considered, android features factored, and the assessment scale used. The authors [50] presented a dynamic analysis approach that collects "second-step behavior cues" to aid application analysis in distinguishing between harmful and benign actions. The features of malicious operations are summarized, and they can be used to categorize them. SSdroid, an analytical prototype, was intended to derive the SSBFs structure of the network. Second step behavior features (SSBFs) have proven successful and valuable in over 9000 activities from benign and hostile programs. Research work has proposed a software library that enables older mobile crowdsourcing applications to boost client security without sacrificing crowdsourced datasets' fundamental value [51]. The research presents Fougere, a decentralized method for transmitting raw data from consumer devices to third-party databases. To examine this contribution, Fougere was deployed in a simulated Android platform by simulating a crowd of fifteen portable devices running multiple variations of MobiPerf and Fougere.

Research explains how to utilize application instrumentation to impose fine-grained usage control privacy limitations, allowing users to regulate how sensitive resources are accessed by applications [52]. The study aimed to give users more control over their mobile devices' privacy, confidentiality, and security, especially when it comes to invasive software behaviors. Rather than providing precise API methods that could allow a program to get private data, this approach provides unambiguous regulations that consider the resource being accessed. The study has examined whether a user's personality is a good predictor of the privacy access they had grant to apps downloaded on their smartphones [53]. The IUIPC questionnaire results and the app permission parameters that were chosen are presented. Deep learning meprivateere utilized to fgrantedst personalized privacy controls for users based on their characteristics and as a result, a unique way for providing permissions to apps is introduced. Research evaluated the privacy and security factor of an Android App in context of how vulnerable it is to leaking end-user personal data and disclosing vulnerabilities [54]. Android static the researchers were collected, and the static code metrics were utilized to estimate the applications' confidentiality risks. The Androrisk tool, which assesses the measure of privacy and security threat of an Android application as determined by an evaluation of Androidvileges and simulation models. The creation of a computational intelligence anti-malware framework (CIantiMF) is proposed [55]. It is built on powerful computational intelligence technologies and runs on the Android operating system. The Android OS was chosen due to its popularity and a large number of vital applications accessible for it. The CIantiMF can determine whether an android application's java classes are benign or malicious. It analyzes network data in order to find Tor-based Botnets. Abstract attack models are proposed to the semantics of various Android attacks are accurately identified [56]. The inter-component communication graph (ICCG) is a new diagram architecture for representing programs' internal management flows and internal component interactions. To discover attacks buried in ICCG, a static searching strategy is proposed and an effective algorithm to search for assaults in ICCG. Experiments have shown that the strategy is viable and effective.

Three new vulnerabilities connected to Web View are revealed in the study exposing new attack surfaces for the most well-known vulnerability related to JavaScript APIs [57]. A static analysis method based on a set of unique inference rules is designed to detect these new forms of vulnerabilities by creating a system that can detect the vulnerability stated in the state-of-the-art. In the study, Babel View was unable to detect three new types of vulnerabilities and was less precise and efficient in detecting previously known flaws. A method was explored, and it was discovered that using description and permission. As a result, a high number of false positives was shown [58]. It was suggested that the app's privacy policy and byte code be used to improve the description and authorization for malware identification. Automatically analyzing privacy policies and performing cross-verification among various types of software artifacts is a difficult task. TAP Verifier is a revolutionary data flow model for assessing privacy policies and developing a novel system to investigate individual software artifacts and do cross verification. Researchers attempted to address the issue by presenting a behavior-based approach to detecting fraudulent Android programs [59]. An application's events and behavioral actions were utilized to create a signature compared to a signature database to detect it. The approach has shown to be effective in detecting malicious activities. It also describes the type of malicious behavior that an app can engage in. The method was shown to be effective since it provides a lot of information on virus interactions that lead to security and privacy issues. Research work was presented, graphs were used to model standard permissions requested by category, and a five-step technique was proposed [60]. It suggests a privacy score relating to risk caution criteria and the similarity of an application to a specified group structure. The threat caution was put to the test in terms of malware identification, and it outperformed similar modern efforts. The suggested study has addressed several current difficulties based on the best parameters and recommends a privacy rating and a threat alert level. The relation between terminal reviews on safety and confidentiality software updates is given [61]. Researchers looked at the influence of user ratings on the security and privacy features of Android applications that are categorized autonomously. Researchers showed that prior security and privacy relevant reviews are a vital component combining exploratory data regression and correlation analysis to estimate confidentiality application changes. The approach may inspire future research that uses user reviews to assess the impact of regulatory changes or modifications to Android's design.

The efficiency of machine learning detectable classifiers fraudulent applications in Android OS is thoroughly examined [62]. The study also has discussed different techniques to estimate the relevancy of the essential features. An Android OS provides a comprehensive assessment in machine learning-based malware detection. The study is based on a more extensive set of data that is open and free to use upon request. Even if only app intrinsic attributes are analyzed, the study confirms that machine learning may be a robust framework for detecting fraudulent apps. AutoPPG is a revolutionary approach that is proposed to make the development of privacy policies for Android applications easier. Based on its description, AutoPPG can recognize the personal data acquired by an API, do static code analysis to characterize its behaviors linked to the user's private information, and then use NLP techniques to generate readable words to describe these behaviors. The AutoPPG trial findings produce accurate privacy policy descriptions and reveal more user-related processes [63]. The study has proposed an essential distinction between harmful and benign actions associated with user intents [64]. Researchers provided an abstraction-based method for extracting and

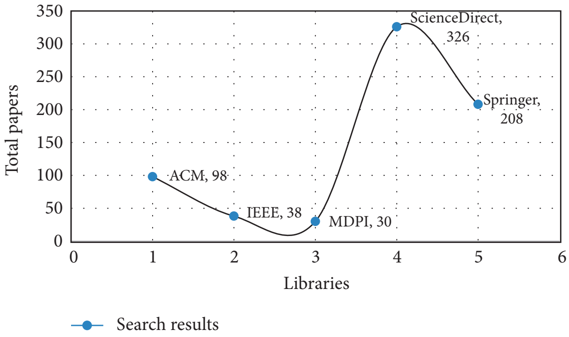

Figure 1: Overall search results in the given libraries.

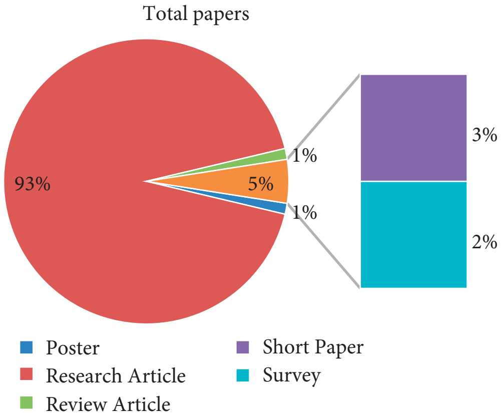

Figure 2: Total contents.

transforming user intention information into a set of essential features that can be utilized to tell the difference between good and bad behavior. IBdroid created and implemented an Android monitor system to carry out this strategy that can track the user interface, user actions, and surveillance behaviors with pinpoint accuracy. Precision and recall are accurately identified in this assessment. A unique method for detecting privacy policy violations caused by the leakage of user input data is provided [65]. Researchers updated the GATOR framework, and a layered object tracking violation detection system was constructed to handle the two technological constraints of infinite mapping and different GUI implementations. The suggested method was tested on three major domains, detecting both solid and weak violations in some of the domains' most popular apps. According to the trial, with the suitable similarity threshold established, this best technique variation may obtain 84 percent on general violation detection.

Research work established a new assessment approach based on analytical hierarchy theory and assessments and descriptions of recent design flaws or data security issues in Android apps [66]. The effort put forth in the project is estimated to make a substantial contribution to the security of Android application data. The study evaluation approach analyzed the data protection of Android apps on different stages and aspects but cannot account for all threats to Android application information security. A VULPIX tool for the characteristics of dynamic analysis is combined to provide a comprehensive identification of privacy vulnerabilities in Android apps [67]. A detailed set of PI data items is also defined, which can be used to compare different PI detection methods. On a collection of Android apps, the consistency of detecting leaked confidential info is assessed using a comprehensive list of data pieces deemed private data. Another study was presented with MPDroid, a method for identifying a combination of simulation process and information retrieval that determines the minimal authorization for a mobile application based on its application description and API utilization [68]. Descriptive minimal permission set iteration method, and for each phone app, static analysis is utilized to determine the actual access. The methodology can also be used to assess the hazards associated with existing Android apps by analyzing their permissions.

5. Analyzing the Existing State-of-the-Art Research Work Based on Popular Libraries

Research works were presented with a new technique for actively defending against fiddling with the code, and reinstalling assault is provided [69]. Whenever downloading an application, the certificate authenticity is checked first, followed by the APP's integrity. If the certificate of authentication is accessed by the application but

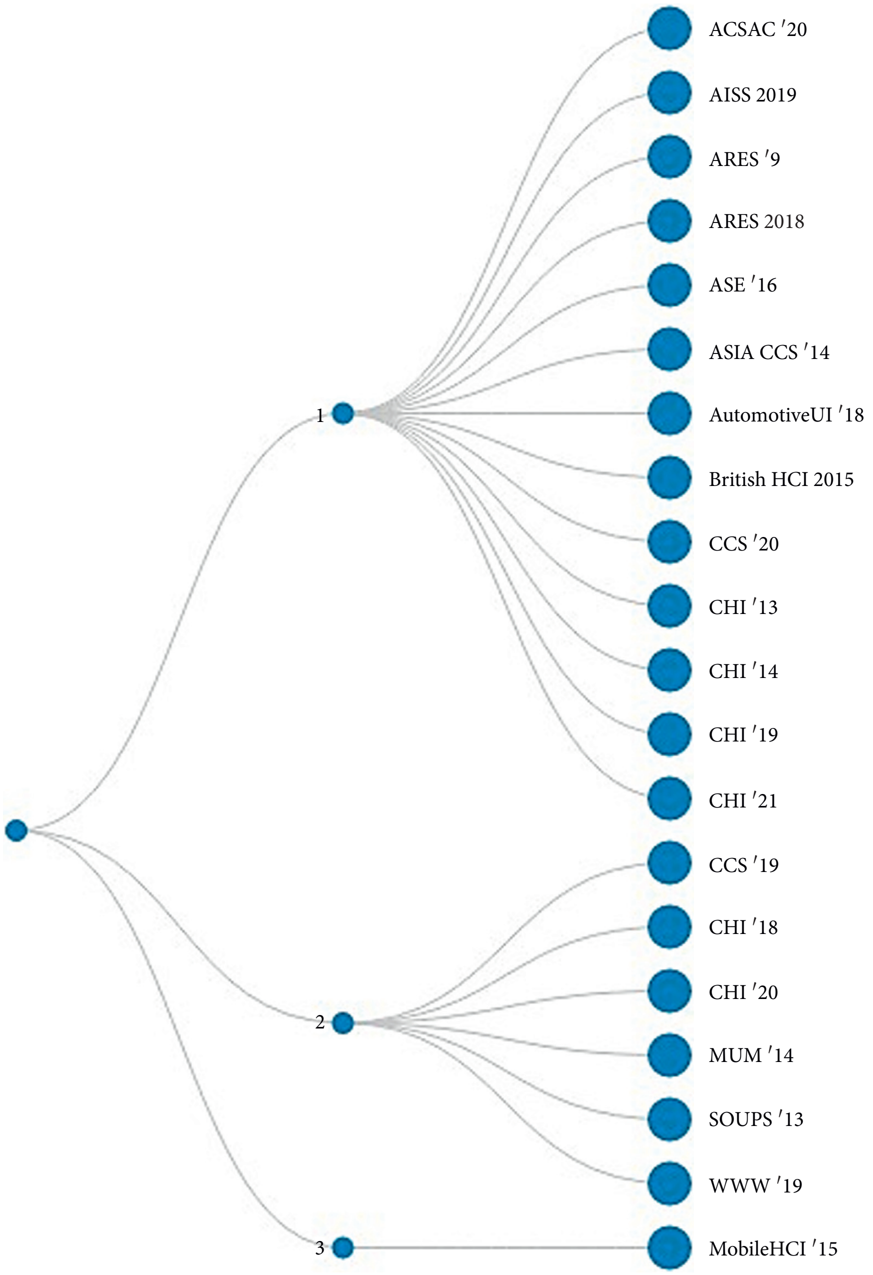

Figure 3: Events of conference.

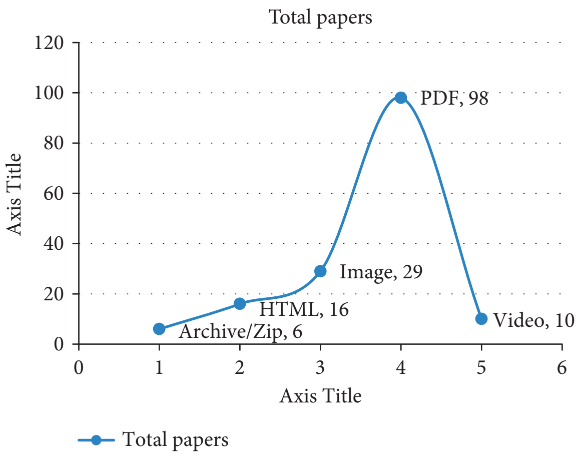

Figure 4: Format of media.

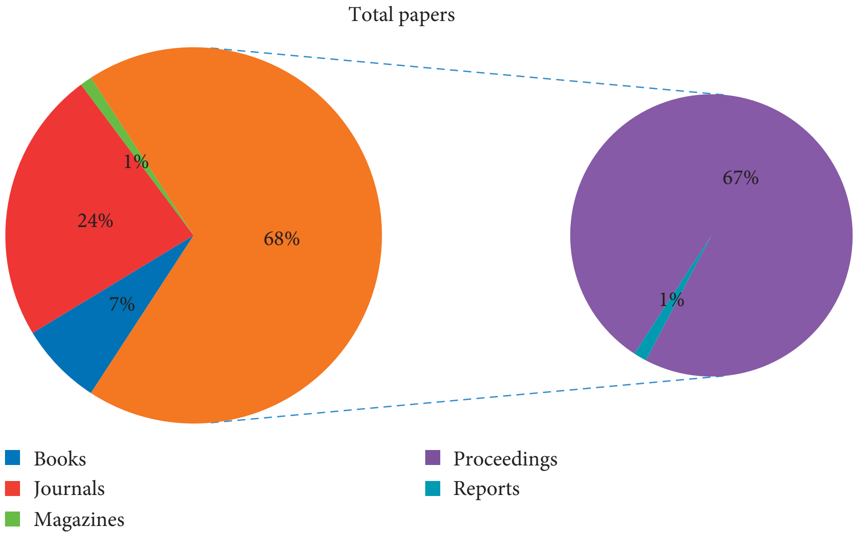

Figure 5: All publications.

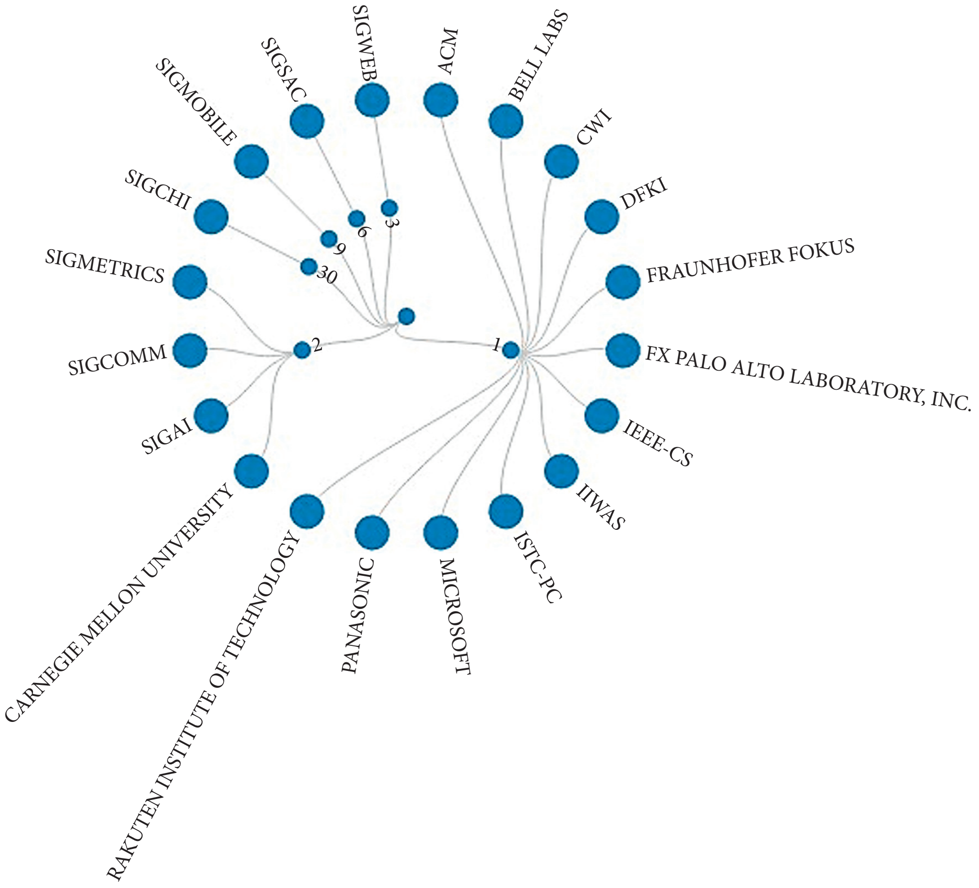

Figure 6: Sponsors of conferences.

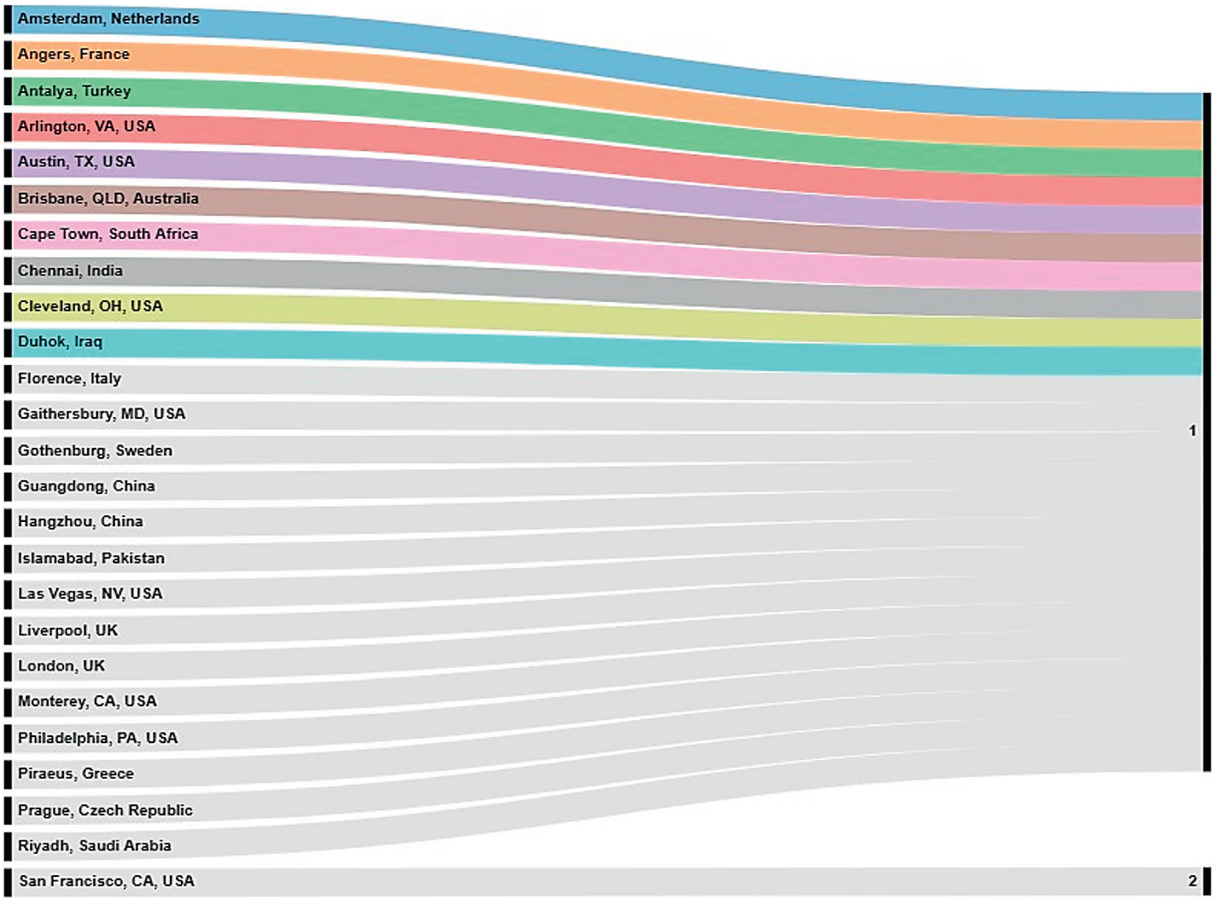

Figure 7: Locations of conferences held.

still has a flaw, the rogue developer can be easily identified. If the attacker updates the Application source and verifies it with unauthentic certificates, the certificate authentication will fail. The root source of code alteration and rebranding assaults is eliminated with this strategy. Another study is presented with documents the vocabulary used in Android app privacy policies to characterize the use of potentially harmful permissions [70]. The semi-automated approach uses NLP and IE approaches to link privacy policies' terminology to dangerous Android permissions. The study yielded over a hundred privacy policy terminology that corresponded to Android problematic permissions. The outcomes of this study serve as the foundation for future research in which the rationales for harmful permissions will be derived automatically from Android app privacy rules. Finally, researchers presented privacy guardian, a preventative policy enforcement system that can prevent malicious attacks in a short period [71]. Rather than being user-defined, the policy rules are based on the behavior model of privacy assaults. The user can choose whether to accept or refuse the procedure. The execution of privacy guardian is described as a finite state machine, and Linear Temporal Logic is used to show its security features. In this article, the effectiveness of privacy guardian was analyzed through the ROC gap, the average accuracy, and the positivity rate.

The first solution for automatically analyzing HTTP(s) predicted on APP protocol behavior for Android apps is presented [72]. Extractocol takes the application binal as input and employs dynamic program analysis and semantic analysis to reconstruct application-specific HTTP-based interactions, resulting in a complete characterization of application protocol behaviors. In-depth testing on both closed source and open source apps demonstrates the accuracy in identifying protocol messages and detailed characterization of app behavior that can be reverse-engineered. A study was presented to characterize and detect a method known as self-hiding behavior [73]. Research and a series of fundamental analyses focusing on SH in Android apps fill the need. According to the findings, the existence of self-concealment in an app is substantially linked to malice. Nonetheless, it has discovered a slew of benign, widely used apps that use masking strategies, implying end consumers and retailers would be benefited from utilizing a method like ours to expose potentially malicious improved user experience by modifying behavior in Android applications. The research aims to develop an excellent way to protect Android app users' private information [74]. A poll of smartphone users was done as part of the study to determine their comprehension of privacy protection. Simultaneously, certain existing approaches are compared to determine their flaws. When the project is completed, the new confidentiality functions provided by MIUI twelve were a

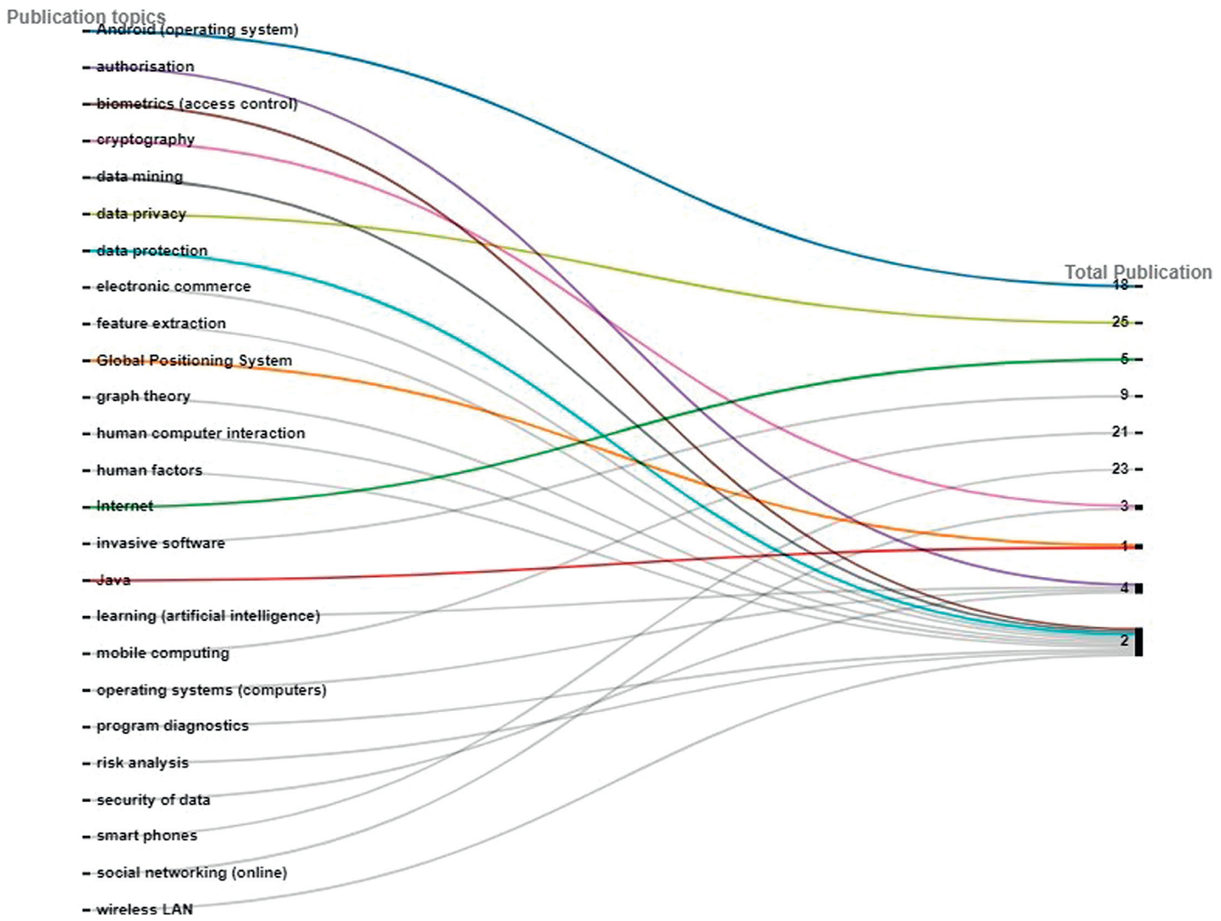

FIGURE 8: Topics of publications.

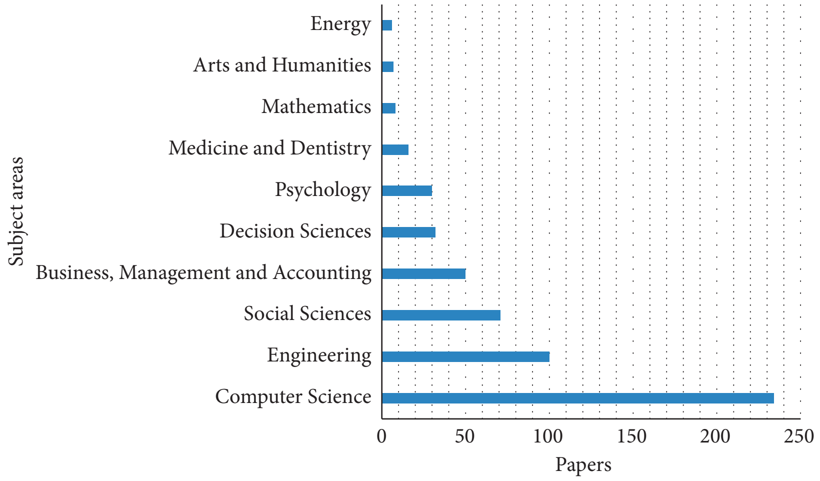

FIGURE 9: Subject areas.

good answer for preventing user secret and limiting app privileges. The terms of service and mobile application privacy rules raise many privacy concerns, which are discussed [75]. To assist designers in the development of applications, guidelines for establishing trust and privacy are offered. The applications' terms of use and privacy policies were examined in light of the guidelines. The Waze application was given special attention because it exemplifies all of the concerns and also demonstrates viable solutions generated through a participatory design session and implementation of the principles developed.

The aim of the proposed study is to present an analysis of the existing literature to show associated materials published in the area. This will help researchers to offer the contents

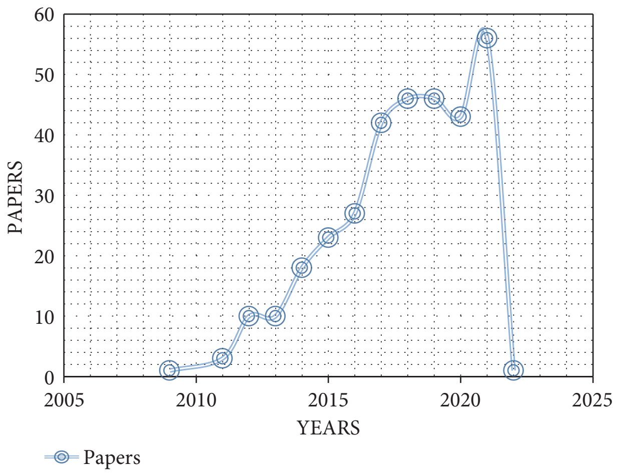

Figure 10: Years of publications.

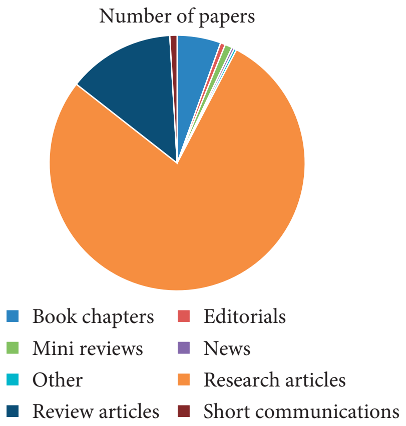

Figure 11: Articles types.

published in the area and then will devise new approaches to overcome challenges. The current study has considered the famous libraries with the search keywords "Influencing," "User's Behavior," "Android," and "Privacy." Details of the libraries are given in the below subsections. Figure 1 describes the overall search results of all the libraries.

5.1. Search Process in the ACM Library. The ACM library was searched for the diverse format of the search process. The analysis derived from the search process is shown in different figures. The total contents in the ACM library are given in Figure 2.

The events of the conference are described in Figure 3.

The media format is shown in Figure 4.

All publications in the given library are shown in Figure 5.

The sponsors of conferences are given in Figure 6. The figure shows that more publications were done in the SIGCHI.

5.2. Search Process in the IEEE Library. The details of the IEEE library were presented in a different format, such as conference locations are shown in Figure 7.

The topics of publications are given in Figure 8.

5.3. Search Process in the ScienceDirect Library. Details of the ScienceDirect library are given with the details of subject areas are given in Figure 9.

The years of publications are shown in Figure 10. The figure shows that there is an increase in the number of publications year-wise.

Details of the article types are given in Figure 11. The figure describes that more publications were published as research articles.

The publications titles with papers are given in Figure 12.

5.4. Search Process in the MDPI Library. Details of the MDPI library are given with the details of subjects areas in Figure 13.

Details of the journals are depicted in Figure 14.

The county details are given in Figure 15.

Figure 16 shows the articles types. This figure shows that more papers were published as journal articles.

5.5. Search Process in the Springer Library. The Springer library details are given below. Figure 17 describes the details of article types in the given library.

Details of the disciplines are shown in Figure 18.

Figure 19 shows the sub-disciplines in the given library.

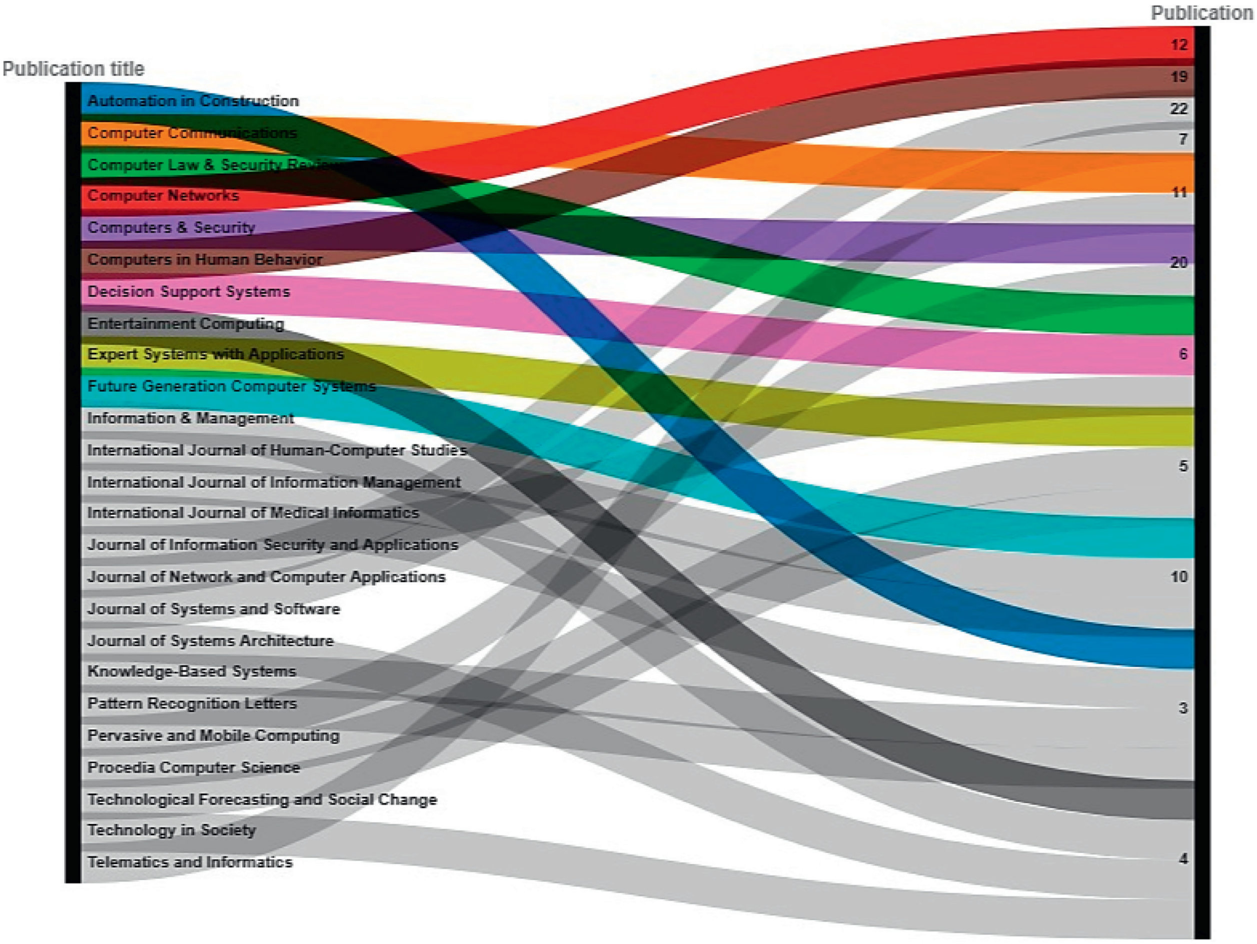

Figure 12: Publications titles.

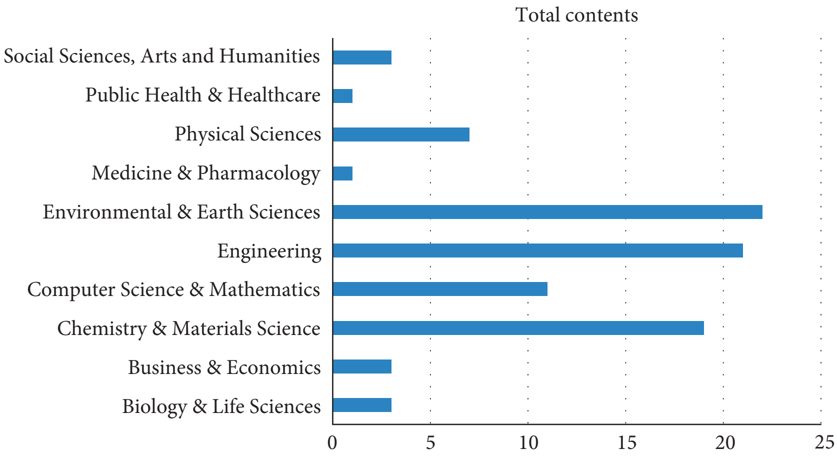

Figure 13: Subject areas.

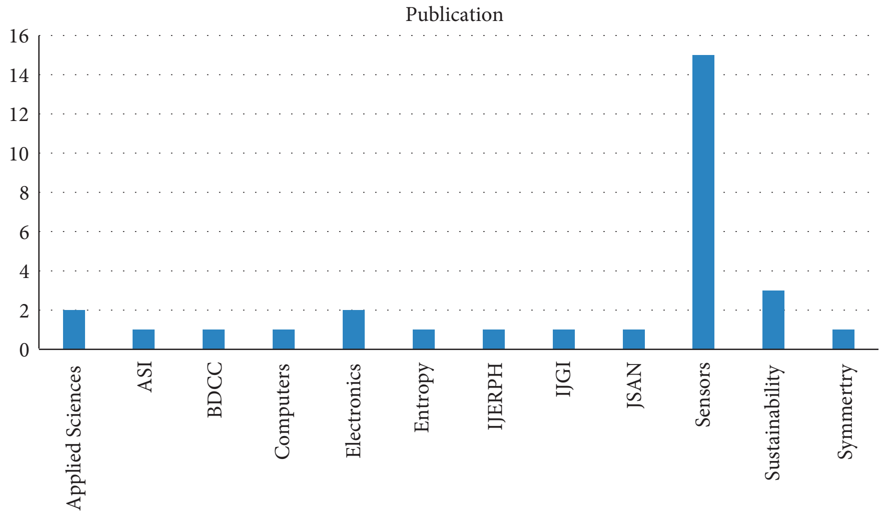

Figure 14: Journals with publications.

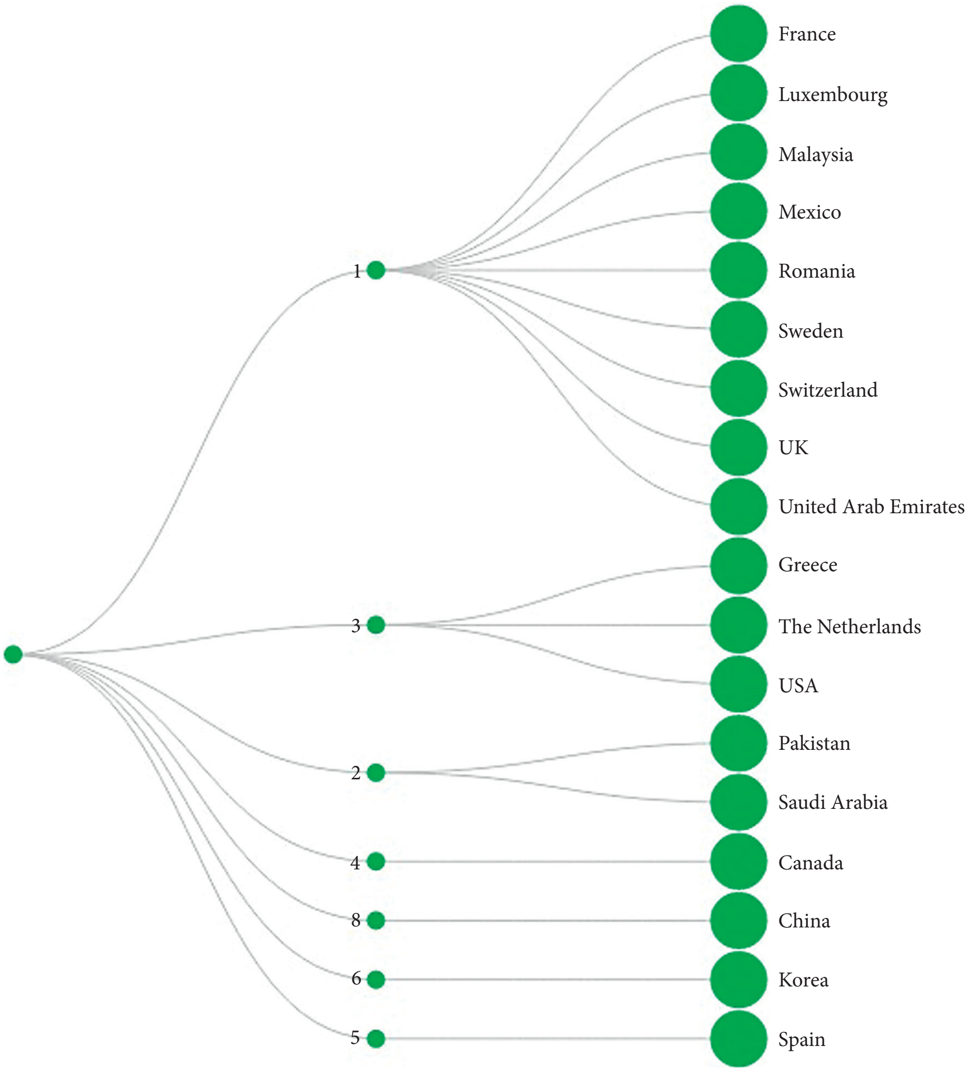

Figure 15: Countries details.

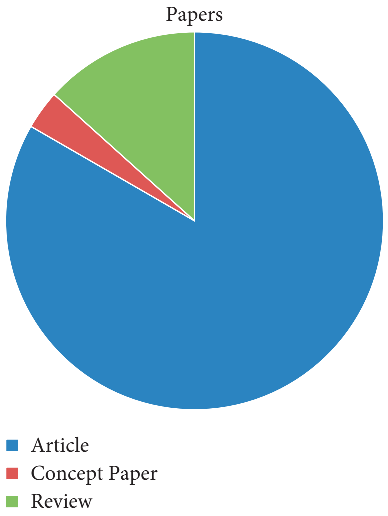

Figure 16: Article types.

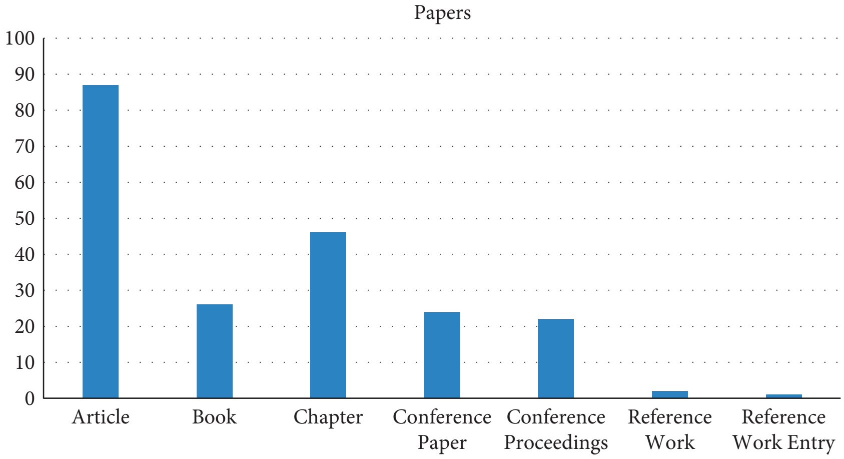

Figure 17: Article types description.

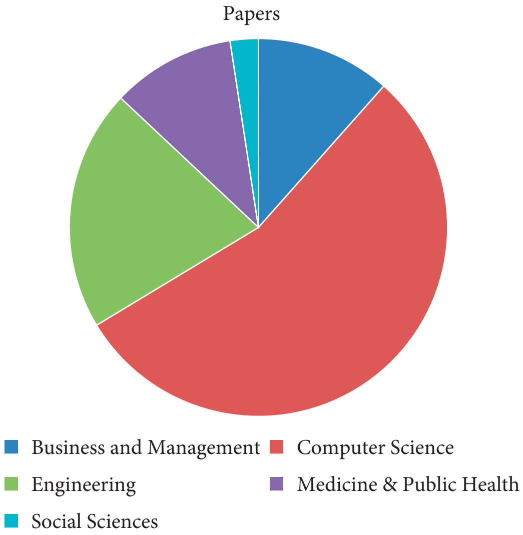

Figure 18: Details of the disciplines.

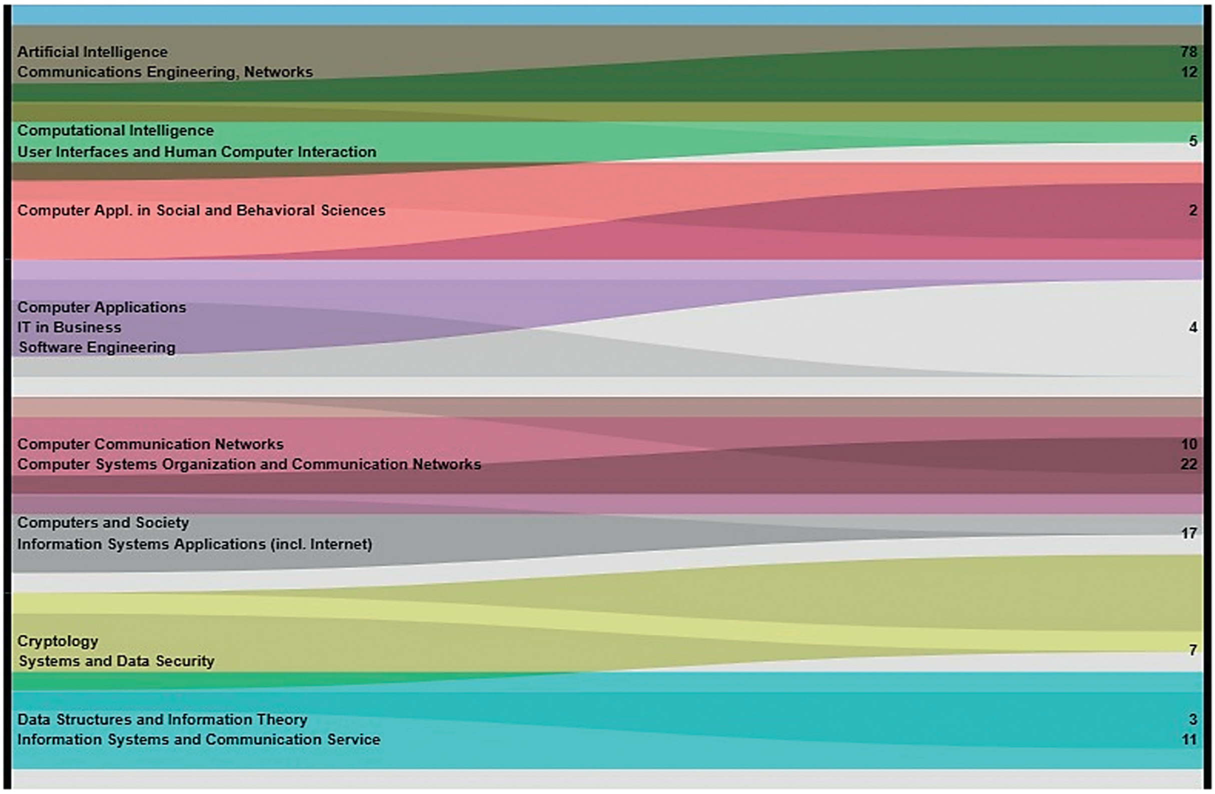

Figure 19: Sub-disciplines.

6. Conclusion

The widespread use and extensive applications of Android in various fields have given birth to new challenges and issues. Mobile applications are easily reachable to normal users of mobile. The mechanism of permission granting is one of the flaws imposed by the developers. Imposing such defects does not permit the user to simply understand the consequences of privacy for conceding permission. Despite conceivable applications for improving the affordability, availability, and effectiveness of delivering various services, it handles sensitive data and information. Risks of security and privacy leakages exist in such data and information. Users are usually unaware of how the data can be managed and used. The existing mobile applications have deficiencies of threats due to design ambiguities. A coherent and comprehensible framework is the dire need of modern industry to facilitate security and privacy solutions to overcome security breaches. This study has offered an overview of the current methodologies, approaches, and tools used for manipulating user behavior regarding the android privacy policy, keeping in view the security and privacy concerns. Numerous prominent libraries were searched, and their search results were analyzed briefly. The search results of these libraries were collected, analyzed, and then presented. This will help researchers in the field formulate novel solutions and overcome diverse challenges.

Data Availability

No data are available.

Conflicts of Interest

The authors declare no conflicts of interest.

References

[1] C. C. Sai, C. S. Prakash, J. Jose, S. C. Mana, and B. K. Samhitha, “Analysing android app privacy using classification algorithm,” in *Proceedings of the 2020 4th International Conference on Trends in Electronics and Informatics (ICOEI)*, pp. 551–555, Tirunelveli, India, 2020.

[2] S. R. Moore, H. Ge, N. Li, and R. W. Proctor, “Cybersecurity for android applications: permissions in android 5 and 6,” *International Journal of Human-Computer Interaction*, vol. 35, no. 7, pp. 630–640, 2019.

[3] A. Hamed and H. K. Ben Ayed, “Privacy risk assessment and users’ awareness for mobile apps permissions,” in *Proceedings of the 2016 IEEE/ACS 13th International Conference of Computer Systems and Applications (AICCSA)*, pp. 1–8, Agadir, Morocco, 2016.

[4] J. Karmazín and P. Očenášek, “The state of near-field communication (NFC) on the android platform,” in *Human Aspects of Information Security, Privacy, and Trust*, T. T. Cham, Ed., pp. 247–254, Springer International Publishing, Berlin, Germany, 2016.

[5] F. Ebrahimi, M. Tushev, and A. Mahmoud, "Mobile app privacy in software engineering research: a systematic mapping study," *Information and Software Technology*, vol. 133, Article ID 106466, 2021.

[6] K. O'Loughlin, M. Neary, E. C. Adkins, and S. M. Schueller, "Reviewing the data security and privacy policies of mobile apps for depression," *Internet Interventions*, vol. 15, pp. 110–115, 2019.

[7] D. Amalfitano, A. R. Fasolino, P. Tramontana, and B. Robbins, "Testing android mobile applications: challenges, strategies, and approaches," in *Advances in Computers*, A. Memon, Ed., vol. 89, pp. 1–52, Elsevier, Amsterdam, Netherlands, 2013.

[8] Z. Alkindi, M. Sarrab, and N. Alzidi, "CUPA: a configurable user privacy approach for android mobile application," in *Proceedings of the 2020 7th IEEE International Conference on Cyber Security and Cloud Computing (CSCloud)/2020 6th IEEE International Conference on Edge Computing and Scalable Cloud (EdgeCom)*, pp. 216–221, New York, NY, USA, 2020.

[9] G. Shrivastava and P. Kumar, "Android application behavioural analysis for data leakage," *Expert Systems*, vol. 38, no. 1, Article ID e12468, 2021.

[10] B. Rashidi, C. Fung, A. Nguyen, and T. Vu, "Android permission recommendation using transitive bayesian inference model," in *Computer Security—ESORICS 2016*, pp. 477–497, Springer International Publishing, Cham, Switzerland, 2016.

[11] N. Wang, B. Zhang, B. Liu, and H. Jin, "Investigating effects of control and ads awareness on android users' privacy behaviors and perceptions," in *Proceedings of the 17th International Conference on Human-Computer Interaction with Mobile Devices and Services*, Copenhagen, Denmark, 2015.

[12] Z. Tang, M. Xue, G. Meng et al., "Securing android applications via edge assistant third-party library detection," *Computers & Security*, vol. 80, pp. 257–272, 2019.

[13] R. Slavin, X. Wang, M. B. Hosseini et al., "Toward a framework for detecting privacy policy violations in android application code," in *Proceedings of the 2016 IEEE/ACM 38th International Conference on Software Engineering (ICSE)*, pp. 25–36, Austin, TX, USA, 2016.

[14] G. Bal, K. Rannenberg, and J. I. Hong, "Styx: privacy risk communication for the android smartphone platform based on apps' data-access behavior patterns," *Computers & Security*, vol. 53, pp. 187–202, 2015.

[15] M. M. Al Sobeihy, "Towards an application-based notion of anomalous privacy behavior in android applications," in *Proceedings of the 1st International Conference on Computer Applications & Information Security (ICCAIS)*, pp. 1–6, Riyadh, Saudi Arabia, 2018.

[16] A. Ngobeni and S. Mhlongo, "Towards enhancing security in android operating systems - android permissions & user unawareness," in *Proceedings of the 2019 2nd International Conference on Computer Applications & Information Security (ICCAIS)*, pp. 1–6, Riyadh, Saudi Arabia, 2019.

[17] Y. Zhou, L. Qi, A. Raake, T. Xu, M. Piekarska, and X. Zhang, "User attitudes and behaviors toward personalized control of privacy settings on smartphones," *Concurrency and Computation: Practice and Experience*, vol. 31, no. 22, Article ID e4884, 2019.

[18] J. Gu, C. Li, D. Lei, and Q. Li, "Combination attack of android applications analysis scheme based on privacy leak," in *Proceedings of the 2016 4th International Conference on Cloud Computing and Intelligence Systems (CCIS)*, pp. 62–66, Beijing, China, 2016.

[19] B. F. Demissie, M. Ceccato, and L. K. Shar, "Security analysis of permission re-delegation vulnerabilities in android apps," *Empirical Software Engineering*, vol. 25, no. 6, pp. 5084–5136, 2020.

[20] G. Dini, F. Martinelli, I. Matteucci, M. Petrocchi, A. Saracino, and D. Sgandurra, "Risk analysis of android applications: a user-centric solution," *Future Generation Computer Systems*, vol. 80, pp. 505–518, 2018.

[21] H. Elahi, A. Castiglione, G. Wang, and O. Geman, "A human-centered artificial intelligence approach for privacy protection of elderly app users in smart cities," *Neurocomputing*, vol. 444, pp. 189–202, 2021.

[22] G. Shrivastava, P. Kumar, D. Gupta, and J. J. P. C. Rodrigues, "Privacy issues of android application permissions: a literature review," *Transactions on Emerging Telecommunications Technologies*, vol. 31, no. 12, Article ID e3773, 2020.

[23] R. S. Gaharwar and R. Gupta, "Vulnerability assessment of android instant messaging application and network intrusion detection prevention systems," *Journal of Statistics and Management Systems*, vol. 23, no. 2, pp. 399–406, 2020.

[24] A. Hamed, H. Kaffel-Ben Ayed, and D. Machfar, "Assessment for android apps permissions a proactive approach toward privacy risk," in *Proceedings of the 2017 13th International Wireless Communications and Mobile Computing Conference (IWCMC)*, pp. 1465–1470, Valencia, Spain, 2017.

[25] M. Hatamian, J. Serna, and K. Rannenberg, "Revealing the unrevealed: mining smartphone users privacy perception on app markets," *Computers & Security*, vol. 83, pp. 332–353, 2019.

[26] S. Havelka, "Typologies of mobile privacy behavior and attitude: a case study comparing German and American library and information science students," *The Serials Librarian*, 2021.

[27] Y. He, C. Wang, G. Xu, W. Lian, H. Xian, and W. Wang, "Privacy-preserving categorization of mobile applications based on large-scale usage data," *Information Sciences*, vol. 514, pp. 557–570, 2020.

[28] Y.-Y. Hong, Y.-P. Wang, and J. Yin, "NativeProtector: protecting android applications by isolating and intercepting third-party native libraries," in *ICT Systems Security and Privacy Protection*, J.-H. Hoepman and S. Katzenbeisser, Eds., Springer International Publishing, Cham, Switzerland, pp. 337–351, 2016.

[29] A. Hoog, "Android device, data, and app security," *Android Forensics*, Elsevier, Amsterdam, Netherlands, pp. 159–194, 2011.

[30] H. Chen, H.-f. Leung, B. Han, and J. Su, "Automatic privacy leakage detection for massive android apps via a novel hybrid approach," in *Proceedings of the 2017 IEEE International Conference on Communications (ICC)*, pp. 1–7, Paris, France, 2017.

[31] Y. Ge, B. Deng, Y. Sun et al., "A comprehensive investigation of user privacy leakage to android applications," in *Proceedings of the 2016 25th International Conference on Computer Communication and Networks (ICCCN)*, pp. 1–6, Waikoloa, HI, USA, 2016.

[32] M. B. Hosseini, "Semantic inference from natural language privacy policies and android code," in *Proceedings of the 2018 26th ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering*, Lake Buena Vista, FL, USA, 2018.

[33] X. Zhu, P. Zhu, and Y. Qiu, "Research on the influence of fear appeals on app users' privacy protection behavior," in

Proceedings of the 2018 International Conference on Information Management & Management Science, Chengdu, China, 2018.

[34] M. Furini, S. Mirri, M. Montangero, and C. Prandi, "Privacy perception and user behavior in the mobile ecosystem," in *Proceedings of the 5th EAI International Conference on Smart Objects and Technologies for Social Good*, Valencia, Spain, 2019.

[35] K. Micinski, D. Votipka, R. Stevens, N. Kofinas, M. L. Mazurek, and J. S. Foster, "User interactions and permission use on android," in *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*, Denver, CO, USA, 2017.

[36] A. Khatoon and P. Corcoran, "Android permission system and user privacy—a review of concept and approaches," in *Proceedings of the 2017 IEEE 7th International Conference on Consumer Electronics—Berlin (ICCE-Berlin)*, pp. 153–158, Berlin, Germany, 2017.

[37] A. Argudo, G. López, and F. Sánchez, "Privacy vulnerability analysis for android applications: a practical approach," in *Proceedings of the 2017 4th International Conference on eDemocracy & eGovernment (ICEDEG)*, pp. 256–260, Quito, Ecuador, 2017.

[38] S. Kumar and R. Shanker, "Your privacy is not so private: unveiling android apps privacy framework from the dark," in *Proceedings of the 2017 International Conference on Next Generation Computing and Information Systems (ICNGCIS)*, pp. 48–53, Jammu, India, 2017.

[39] L. Verderame, D. Caputo, A. Romdhana, and A. Merlo, "On the (un)reliability of privacy policies in android apps," in *Proceedings of the 2020 International Joint Conference on Neural Networks (IJCNN)*, pp. 1–9, Glasgow, UK, 2020.

[40] P. Singh, S. Singh, and P. Tiwari, "Discovering persuaded risk of permission in android applications for malicious application detection," in *Proceedings of the 2016 International Conference on Inventive Computation Technologies (ICICT)*, vol. 3, pp. 1–5, Coimbatore, India, 2016.

[41] D. Geneiatakis, I. N. Fovino, I. Kounelis, and P. Stirparo, "A permission verification approach for android mobile applications," *Computers & Security*, vol. 49, pp. 192–205, 2015.

[42] V. M. Wottrich, E. A. van Reijmersdal, and E. G. Smit, "The privacy trade-off for mobile app downloads: the roles of app value, intrusiveness, and privacy concerns," *Decision Support Systems*, vol. 106, pp. 44–52, 2018.

[43] X. Wang, W. Wang, Y. He, J. Liu, Z. Han, and X. Zhang, "Characterizing android apps' behavior for effective detection of malapps at large scale," *Future Generation Computer Systems*, vol. 75, pp. 30–45, 2017.

[44] D. Schreckling, J. Köstler, and M. Schaff, "Kynoid: real-time enforcement of fine-grained, user-defined, and data-centric security policies for android," *Information Security Technical Report*, vol. 17, no. 3, pp. 71–80, 2013.

[45] V. Sihag, M. Vardhan, and P. Singh, "A survey of android application and malware hardening," *Computer Science Review*, vol. 39, Article ID 100365, 2021.

[46] V. M. Wottrich, E. A. Reijmersdal, and E. G. Smit, "App users unwittingly in the spotlight: a model of privacy protection in mobile apps," *Journal of Consumer Affairs*, vol. 53, no. 3, pp. 1056–1083, 2019.

[47] Y. Kouraogo, K. Zkik, E. J. El Idrissi Noreddine, and G. Orhanou, "Attacks on android banking applications," in *Proceedings of the 2016 International Conference on Engineering & MIS (ICEMIS)*, pp. 1–6, Agadir, Morocco, 2016.

[48] H. Li, W. Liu, B. Wang, and W. Zhang, "Detection and autoprotection of cache file privacy leakage for mobile social networking applications in android," in *Human Aspects of Information Security, Privacy and Trust*, T. Tryfonas, Ed., Springer International Publishing, Cham, Switzerland, pp. 703–721, 2017.

[49] L. Li, T. F. Bissyandé, M. Papadakis et al., "Static analysis of android apps: a systematic literature review," *Information and Software Technology*, vol. 88, pp. 67–95, 2017.

[50] P. Li, J. Fu, C. Xu, B. Cheng, and H. Zhang, "Differentiating malicious and benign android app operations using second-step behavior features," *Chinese Journal of Electronics*, vol. 28, no. 5, pp. 944–952, 2019.

[51] L. Meftah, R. Rouvoy, and I. Chrisment, "Empowering mobile crowdsourcing apps with user privacy control," *Journal of Parallel and Distributed Computing*, vol. 147, pp. 1–15, 2021.

[52] R. Neisse, G. Steri, D. Geneiatakis, and I. Nai Fovino, "A privacy enforcing framework for android applications," *Computers & Security*, vol. 62, pp. 257–277, 2016.

[53] F. Raber and A. Krueger, "Towards understanding the influence of personality on mobile app permission settings," in *Human-Computer Interaction—INTERACT 2017*, pp. 62–82, Springer International Publishing, Cham, Switzerland, 2017.

[54] A. Rahman, P. Pradhan, A. Partho, and L. Williams, "Predicting android application security and privacy risk with static code metrics," in *Proceedings of the 2017 IEEE/ACM 4th International Conference on Mobile Software Engineering and Systems (MOBILESoft)*, Buenos Aires, Argentina, 2017.

[55] K. Demertzis and L. Iliadis, "Computational intelligence antimalware framework for android OS," *Vietnam Journal of Computer Science*, vol. 4, no. 4, pp. 245–259, 2017.

[56] G. Meng, R. Feng, G. Bai, K. Chen, and Y. Liu, "DroidEcho: an in-depth dissection of malicious behaviors in android applications," *Cybersecurity*, vol. 1, no. 1, p. 4, 2018.

[57] M. A. El-Zawawy, E. Losiouk, and M. Conti, "Vulnerabilities in android webview objects: still not the end!" *Computers & Security*, vol. 109, Article ID 102395, 2021.

[58] L. Yu, X. Luo, C. Qian, and S. Wang, "Revisiting the description-to-behavior fidelity in android applications," in *Proceedings of the 2016 IEEE 23rd International Conference on Software Analysis, Evolution, and Reengineering (SANER)*, vol. 1, Osaka, Japan, 2016.

[59] V. Sihag, A. Swami, M. Vardhan, and P. Singh, "Signature based malicious behavior detection in android," in *Computing Science, Communication and Security*, N. Chaubey, S. Parikh, and K. Amin, Eds., Springer, Singapore, pp. 251–262, 2020.

[60] K. Sokolova, C. Perez, and M. Lemercier, "Android application classification and anomaly detection with graph-based permission patterns," *Decision Support Systems*, vol. 93, pp. 62–76, 2017.

[61] D. C. Nguyen, E. Derr, M. Backes, and S. Bugiel, "Short text, large effect: measuring the impact of user reviews on android app security & privacy," in *Proceedings of the 2019 IEEE Symposium on Security and Privacy (SP)*, San Francisco, CA, USA, 2019.

[62] V. Syrris and D. Geneiatakis, "On machine learning effectiveness for malware detection in android OS using static analysis data," *Journal of Information Security and Applications*, vol. 59, Article ID 102794, 2021.

[63] L. Yu, T. Zhang, X. Luo, L. Xue, and H. Chang, "Toward automatically generating privacy policy for android apps," *IEEE Transactions on Information Forensics and Security*, vol. 12, no. 4, pp. 865–880, 2017.

[64] J. Fu, P. Li, Y. Lin, and S. Ding, "Android app malicious behavior detection based on user intention," in *Proceedings of the 2016 IEEE Trustcom/BigDataSE/ISPA*, Tianjin, China, 2016.

[65] X. Wang, X. Qin, M. B. Hosseini, R. Slavin, T. D. Breaux, and J. Niu, "GUILeak: tracing privacy policy claims on user input data for android applications," in *Proceedings of the 2018 IEEE/ACM 40th International Conference on Software Engineering (ICSE)*, Gothenburg, Sweden, 2018.

[66] Y. Wan, G. Wang, and X. Feng, "An evaluation model for information security of android application based on analytic hierarchy process," in *Proceedings of the 2016 World Automation Congress (WAC)*, Rio Grande, PR, USA, 2016.

[67] N. Wongwiwatchai, P. Pongkham, and K. Sripanidkulchai, "Comprehensive detection of vulnerable personal information leaks in android applications," in *Proceedings of the IEEE INFOCOM 2020—IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*, Toronto, Canada, 2020.

[68] J. Xiao, S. Chen, Q. He, Z. Feng, and X. Xue, "An android application risk evaluation framework based on minimum permission set identification," *Journal of Systems and Software*, vol. 163, Article ID 110533, 2020.

[69] X. Zhang and R. Zhang, "A solution of code authentication on android," in *Information and Communications Security*, pp. 356–362, Springer International Publishing, Cham, Switzerland, 2016.

[70] R. Baalous and R. Poet, "How dangerous permissions are described in android apps' privacy policies?" in *Proceedings of the 11th International Conference on Security of Information and Networks*, Cardiff, UK, 2018.

[71] D. Zhu, H. Jin, Y. Yang, and W. Chen, "Privacy guardian: preventive policy enforcement against privacy malware on android," in *Proceedings of the 6th International Conference on Information Engineering*, Dalian, China, 2017.

[72] J. Kim, H. Choi, H. Namkung et al., "Enabling automatic protocol behavior analysis for android applications," in *Proceedings of the 12th International on Conference on emerging Networking Experiments and Technologies*, Irvine, CA, USA, 2016.

[73] Z. Shan, I. Neamtiu, and R. Samuel, "Self-hiding behavior in android apps: detection and characterization," in *Proceedings of the 40th International Conference on Software Engineering*, Gothenburg, Sweden, 2018.

[74] J. Wang and R. Poet, "Permission management and user privacy based on android: permission management on android," in *Proceedings of the 13th International Conference on Security of Information and Networks*, Merkez, Turkey, 2020.

[75] E. A. Yamauchi, P. C. d. Souza, and D. P. S. Junior, "Prominent issues for privacy establishment in privacy policies of mobile apps," in *Proceedings of the 15th Brazilian Symposium on Human Factors in Computing Systems*, São Paulo, Brazil, 2016.