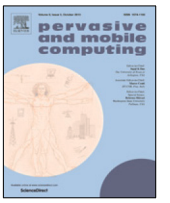
Pervasive and Mobile Computing 92 (2023) 101786

Contents lists available at ScienceDirect

Pervasive and Mobile Computing

journal homepage: www.elsevier.com/locate/pmc



Studying users' perception of IoT mobile companion apps

Gian Luca Scoccia [a,*], Romina Eramo [b], Marco Autili [a]

[a] DISIM, University of L'Aquila, L'Aquila, Italy
[b] University of Teramo, Teramo, Italy

ARTICLE INFO

Article history:
Received 16 September 2022
Received in revised form 4 February 2023
Accepted 2 April 2023
Available online 10 April 2023

Dataset link: https://bit.ly/companionApps

Keywords:
Apps
IoT
Android
iOS
Opinion mining
Review analysis

ABSTRACT

Internet of Things (IoT) products provide over-the-net capabilities such as remote activation, monitoring, and notifications. An associated mobile app is often provided for more convenient usage of these capabilities. The perceived quality of these *companion apps* can impact the success of the IoT product. We investigate the perceived quality and prominent issues of smart-home IoT mobile companion apps with the aim of deriving insights to: (i) provide guidance to end users interested in adopting IoT products; (ii) inform companion app developers and IoT producers about characteristics frequently criticized by users; (iii) highlight open research directions. We employ a mixed-methods approach, analyzing both quantitative and qualitative data. We assess the perceived quality of companion apps by quantitatively analyzing the star rating and the sentiment of 1,347,799 Android and 48,498 iOS user reviews. We identify the prominent issues that afflict companion apps by performing a qualitative manual analysis of 1,000 sampled reviews. Our analysis shows that users' judgment has not improved over the years. A variety of functional and non-functional issues persist, such as difficulties in pairing with the device, software flakiness, poor user interfaces, and presence of issues of a socio-technical impact. Our study highlights several aspects of companion apps that require improvement in order to meet user expectations and identifies future directions.

1. Introduction

Internet of Things (IoT) devices have become part of the daily lives of billions of people. Approximately 500 billion devices are expected to embrace sensors and be associated with the Internet by 2030, becoming a necessary ecosystem in which data, processes, human beings, things and the Internet are associated with each other [1]. These products provide over-the-net capabilities such as remote activation, monitoring, and notifications. An associated mobile app is often provided for a more convenient usage of these capabilities. Hereafter, we will refer to these applications as *companion apps*. Examples are in the field of smart homes, gaming, smartwatches, and sport devices.

IoT is among the most publicized technologies that could change the way businesses operate. The hype around the IoT makes it an essential topic for a business strategy that combines emerging trends and digital transformations. However, the lack of mature development in companion apps can have a big impact on the success of IoT devices. Companion apps, in fact, provide insights into the various aspects of the IoT devices themselves. While several studies are aimed at security and privacy issues [2,3], this work shows that the aspects to be considered to achieve the quality of the services offered by the apps are wider and deserve attention.

* Corresponding author.
E-mail addresses: gianluca.scoccia@univaq.it (G.L. Scoccia), reramo@unite.it (R. Eramo), marco.autili@univaq.it (M. Autili).

https://doi.org/10.1016/j.pmcj.2023.101786

In this paper, we investigate how end users perceive the quality of smart-home IoT mobile companion apps, with the ultimate goal of identifying prominent issues and possible points of improvement. For this purpose, we conducted an empirical study by employing a mixed-methods approach, analyzing both quantitative and qualitative data. As first, we considered 1,347,799 Android and 48,498 iOS user reviews to assess the perceived quality of companion apps by quantitatively analyzing the star rating and the sentiment. We then identified the prominent issues that afflict these apps by performing a qualitative manual analysis of 1,000 sampled reviews; in particular, two experienced researcher manually analyzed the extracted samples independently and categorized the main concerns expressed by end users into different categories. Finally, by analyzing the achieved classification, we identified a number of prominent issues that lead to points for improvement. We discussed each point in details and provide insights for future research and for improving the perceived quality of these apps.

The main contributions of this paper are the following:

- a taxonomy of the main concerns expressed by users in reviews about the Android and iOS smart-home companion apps' quality;
- empirical results about users' perception of smart-home companion apps;
- the identification of a number of potential issues on companion mobile apps quality;
- a discussion about some open research directions.

The target audience of this paper is composed of end users, companion apps developers and IoT producers, and researchers. We support end users interested in adopting IoT products by providing guidance about general aspects and issues shared among the several apps. We support developers and IoT producers by informing them about characteristics of their products frequently criticized by users and by providing a set of actionable and evidence-based insights. We support researchers by informing them on the state of companion apps, providing a classification framework for investigating on arbitrary aspects of companion app user reviews, and discussing open research directions. To allow for independent verification and replication of the performed study, we make publicly available a replication package containing the collected data and all the code developed for data preparation and analysis.[1]

The remainder of the paper is structured as follows. Section 2 describes related work. Section 3 describes the design of our study. Section 4 presents the main results, that are then discussed in Section 5. Section 6 discusses the threats to the validity of our study, Section 7 closes the paper.

2. Related work

In this section, we discuss work related to our study by covering three main topics, i.e., literature about mobile app review analysis, Internet of Things systems, and companion apps.

2.1. Mobile app review analysis

In the literature, a vast amount of research works have been produced to study what useful information might be found in app reviews, how it can be extracted, and how it can be used to facilitate software engineering activities [4,5]. Studies have found that a wide variety of topics is discussed by users in app reviews, including app features [6–9], bug reports [7,10], requirements [11–13], and updates [14–17].

More tightly related to our work are opinion mining studies that analyzed user reviews to discover and understand the issues experienced by users in specific domains. Williams and colleagues [18] present a case study focused on anonymous social networking platforms. Conducting a qualitative analysis of user reviews, they identified seven main concerns experienced by users and their impact on the core features of applications in this domain. Voskobojnikov et al. [19] identified and analyzed user experience issues found in app reviews of mobile cryptocurrency wallets. They found that both new and experienced users struggle with general and domain-specific issues that might result not only in frustration and disengagement but also in dangerous errors and irreversible monetary losses. Mujahid and colleagues mine user reviews in order to understand the user complaints of wearable apps, i.e., software designed to be executed on smartwatches and fitness trackers [20,21]. Their findings indicate that the more frequent complaints are about functional errors, cost, and lack of functionality. However, they found that the more negatively impacting complaints are related to installation problems, device compatibility, and privacy and ethical issues. Garousi et al. [22] analyzed user reviews of nine European contact-tracing apps. Their analysis evidences that for all considered applications design and user experience issues are frequently reported by users. However, differences in quality between nations was observed.

[1] https://bit.ly/companionApps

2.2. Internet of Things

The vision of the Internet of Things is to embed communication capabilities within a highly distributed, ubiquitous and dense heterogeneous devices network to enable new intelligent applications and services [1]. Nowadays, IoT devices are commercially available and have become part of the daily lives of billions of people [1,2]. These new devices pose several technical, organizational, and social challenges [1,23,24].

Limited empirical evidence exists on the challenges that developers face while programming IoT systems software. Makhshari and colleagues [24] performed a systematic study of bugs and challenges that IoT developers face in practice, analyzing 5,565 bug reports from 91 IoT project repositories and validated by interviewing 194 IoT developers. They report difficulties related to immature testing tools, lack of device-level monitoring, and the fragmentation of the IoT ecosystem. Corno [25] et al. investigated the challenges faced by less experienced IoT developers. Their results highlight that the major challenges faced by these developers were due to a lack of well-structured documentation, the complexity inherent to the interplay of the subsystems, and the integration with third-party services. Srisopha et al. [26] conducted an exploratory study to evaluate whether user reviews of commercial IoT products can be an useful source of information for software developers and maintainers of these product. After analyzing 7,198 reviews from 6 commercial IoT products, they report that a sufficient quantity of software related information exists in these reviews. Our study contributes to the construction of a body of empirical evidence, by providing a taxonomy of issues that frequently arise *in-the-wild*.

Usability, adoption, and user acceptance of IoT devices have also been investigated. Oliveira [27] and colleagues conducted a field study in which they interviewed the residents of 19 households, prior to the installation of a smart heating management system and after living with the technology for one year. Comparing the two interview sets, they found that many initial expectations were met but unforeseen issues arose, especially regarding the usability and the effort required to configure the smart devices. Zheng et al. [28] conducted 11 semi-structured interviews with smart homeowners, investigating their reasons for purchasing IoT devices, perceptions of smart home privacy risks, and actions taken to protect their privacy. They highlight several recurring themes in collected responses, among which are the fact that users trust IoT device manufacturers to protect their privacy but do not verify that these protections are in place. Our study complements the ones mentioned above by providing quantitative evidence on the quality of devices perceived by device owners, collected from app stores.

2.3. Companion apps

At the time of writing, companion apps have been studied in the literature exclusively from a security standpoint. Wang and colleagues [2] have been the first to investigate the security of these applications and associated IoT devices. Using a suite of program analysis techniques, the authors analyzed 2,081 Android mobile smart-home companion apps to discover potential security vulnerabilities in over 4,700 IoT devices, leveraging the intuition that software and hardware components are often reused across devices. Their approach successfully identified 324 devices from 73 different vendors likely to be vulnerable to a set of security issues. A follow-up study by Mohanty et al. [3] proposes HybriDiagnostics, a vulnerability assessment framework to uncover security issues in smart-home companion apps developed using hybrid app development frameworks. The authors uncover nine security issues found in popular hybrid frameworks and demonstrate their exploitability in a smart home environment. In our work, we focus on previously unconsidered aspects of companion apps, investigating their perceived quality and prominent issues experienced by users.

3. Study design

This section describes the design of our study. In order to perform an objective and replicable study, we followed the guidelines on empirical software engineering outlined in [29,30].

3.1. Goal and research questions

The *goal* of our study is to investigate the quality and prominent issues of smart-home IoT companion apps, as perceived by users. By analyzing reviews and store metadata of apps, we aim to derive insights to (i) provide guidance to users interested in using IoT products; (ii) inform companion app developers and IoT producers about characteristics of their products on which they should focus their efforts, as these are frequently criticized by users; and (iii) highlight open research directions for researchers. The *context* of our study is the one of real-world Android and iOS smart-home companion apps available on the respective app stores.

To achieve this goal, we define the following research questions:

RQ1 How has the perceived quality of smart-home IoT companion apps evolved over time?
RQ2 Which are the prominent issues of smart-home IoT companion apps perceived by users?

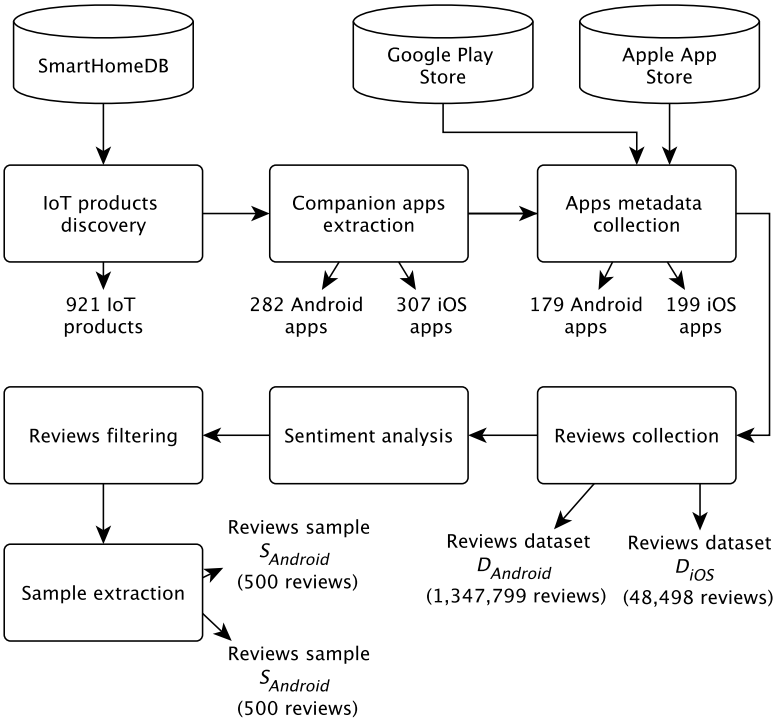


Fig. 1. Data collection process.

By answering RQ1, we investigate what is the opinion of users on smart-home companion apps and, by proxy, on associated IoT products. Moreover, we examine how the opinion has evolved over time, given the potential greater maturity of newer-generation IoT devices. RQ2 instead aims to identify and categorize the main issues currently faced by users, to evidence the most critical aspects of smart-home companion apps and associated devices that require improvement.

In order to answer RQ1, we rely on two metrics to estimate the perceived quality of companion apps: the review *star rating* and the review *sentiment*. The former is a score from one to five given by the review author to the mobile app. The latter is a categorization of opinions expressed in the reviews as negative, neutral, or positive judgments. Both metrics, and the process used to extract them, are described in detail in Section 3.2. Correspondingly, to answer RQ2, there is a need for an assessment of the main issues faced by users of companion apps. This is obtained as the result of a manual analysis procedure, described in Section 3.3.

3.2. Data collection

Fig. 1 summarizes the data collection process employed in our study. The data collection was started in May 2021 and completed in December of the same year. In the following, we describe each step in detail.

3.2.1. IoT products discovery

As for the initial step, we identified a relevant number of smart home IoT products that are controllable by an associated companion app. Following a procedure similar to the one employed by Wang and colleagues [2], we collected information about smart-home IoT devices from `SmartHomeDB`,[2] an open community-supported smart home devices database. Using the offered controls, we queried for those products that are equipped with either an Android or an iOS (at the time of writing, the two most popular mobile operating systems [31]) companion app. This led to the identification of 921 distinct products as of the 24th of May 2021.

3.2.2. Companion apps extraction

Using an ad-hoc script, we iterated over the description page of each product identified in the previous step and extracted the companion app unique identifier (i.e., the package name for Android apps or the app id for iOS ones). This is possible since companion apps cannot be installed directly from SmartHomeDB; rather, the description page of each

2 https://www.smarthomedb.com/

Table 1
Descriptive statistics of collected applications.

Android						
Metric	Min	Max	Median	Mean	SD	IQR
Rating	1.23	4.84	3.21	3.2	0.83	1.29
Reviews	1	278,588	494	8,925.55	31,812.38	2,697
Installations[a]	1k	100M	100k	3,023k	12,811k	490k
iOS						
Metric	Min	Max	Median	Mean	SD	IQR
Rating	1	5	2.94	3.12	1.19	2.34
Reviews	1	3,169,928	187	54,242.59	277,655.1	2,002.5
Installations[b]	–	–	–	–	–	–

SD = standard deviation, IQR = inter-quartile range.
[a]Google Play does not provide the precise number of installations, but only a range (i.e., 100–1000). We conservatively adopted the bottom of the range.
[b]The Apple App store does not publish the number of app installations.

product provides links to the companion app's pages on the respective app stores (i.e., the Google Play store for Android, and the Apple App Store for iOS). Each link embeds the companion app unique identifier, which can be easily extracted by parsing the page `HTML` source code. This allowed us to identify 282 Android and 307 iOS companion apps. We observed that the main reason why companion apps are lower in number than IoT products is that most brands provide a single app to control all products in a line, thus removing the need of a distinct app for each product.

3.2.3. Apps metadata collection

We collected application metadata from the app store page for the IoT companion apps identified in the previous step, leveraging two open-source tools.[3,4] For each application, we collected the application star rating (i.e., a score from one to five that averages the scores given by users), and the application's total number of reviews. Additionally, for Android apps only, we collected the number of application installations. This information is unavailable on the store for iOS apps. During this step, we found that some of the apps were no longer available on the respective app store (i.e., Google Play for Android apps, App Store for iOS ones). This can happen if the app developer decides to remove the app from the store or if the app is found to be in violation of the store policies. Hence, these apps were excluded, leaving us with a final amount of 179 and 199 Android and iOS apps, respectively. Descriptive statistics for the final set of apps included in the study are provided in Table 1.

3.2.4. User reviews collection

For the companion apps surviving the previous step, we collected the user reviews published on the app stores. During this step, one difference among the two platforms was encountered: while the Google Play Store publishes all the reviews written by users, the Apple App Store only provides at most five hundred reviews on the store page of each application. For this reason, we were able to collect all the published reviews for Android apps but only up to five hundred reviews for each iOS app in our dataset, leading to a total amount of 1,597,673 Android reviews and 48,981 iOS ones. The reviews collection was carried out employing the open-source tools used in the previous step.

In order to ensure quality of our dataset, we filtered out all reviews written in a non-English language. For this purpose, we employed the `LangDetect` language identification library [32]. After this step, a total of 1,347,799 Android and 48,498 iOS reviews remained in our dataset. For Android (iOS) apps, the earliest review collected is dated 9 December 2010 (13th July 2008), while the latest one is dated 24th of May 2021 (20th December 2021). Hereafter, we will refer to the datasets of English Android and iOS reviews as $D_{Android}$ and D_{iOS}, respectively. For each review, in addition to its text, we collected the metadata associated with it, i.e., the user-given rating associated with the review and the publishing date.

3.2.5. Sentiment analysis

We enriched our dataset by performing *sentiment analysis* on $D_{Android}$ and D_{iOS}. Sentiment analysis [33] is a frequently used opinion mining technique whose goal is to identify and extract affective states and subjective information reported in sentences. In its most common usage scenario, sentiment analysis is used to classify written opinions as negative, neutral, or positive. In our study, we adopt the `VADER` [34] sentiment analysis tool, selected among other possibilities (e.g., SentiStrength [35] or Stanford CoreNLP [36]) as it employs a rule-based approach specialized for short texts, such as user reviews. In a comparison of 24 sentiment analysis methods, evaluated over multiple domains such as social network comments and product reviews, `VADER` has shown to be the most consistently well performing solution [37].

[3] https://github.com/facundoolano/google-play-scraper
[4] https://github.com/facundoolano/app-store-scraper

VADER computes a normalized weighted composite score (compound score) by summing the valence scores of each word in the lexicon, adjusted to the grammatical and syntactic rules, then normalized so to fall in range -1 (most negative) and $+1$ (most positive). We adopted the recommended thresholds for the compound score, used in VADER's [34] own evaluation, which consider an extracted sentiment as *positive* if its compound score is ≥ 0.05, *negative* if it is ≤ -0.05, and *neutral* if between the two thresholds. It is worth to note that we compute the sentiment for all reviews in $D_{Android}$ and D_{iOS} without preemptively discarding short or low-quality reviews. Indeed, these reviews might be uninformative while still expressing a polarized opinion about the app (e.g., *"Terrible app!"*). Hence, discarding them can potentially introduce a bias in our analysis. Moreover, VADER rule-based extraction engine is particularly suited to deal with these difficult instances [34].

3.2.6. Reviews filtering

In order to improve the relevance of our datasets $D_{Android}$ and D_{iOS}, before performing the subsequent manual analysis steps, we applied two reviews filtering criteria described in the following.

As mentioned, we performed a manual analysis with the aim to build up a ground truth for future automation of the analysis process. Since, to the best of our knowledge, this is the first work that studies the services offered by the companion apps in a general way, we decided not to use automated tools, but to rely on manual analysis. However, manual analysis is time-consuming and requires a vast human effort [38]. Thus, we decided to focus on the reviews that have more potential in terms of relevance and timeliness. As proved in studies dealing with the prioritization of user reviews [39], short reviews are typically categorized as non-informative, while those containing potentially useful feedback are longer and more structured. We filtered out those reviews with lengths less than five words, as these reviews are likely to contain generic praises (e.g., "Good app!") or complaints (e.g., "Doesn't work".) from which it is not possible to extract useful insights. Secondly, we removed all reviews published prior to the year 2019. IoT is rapidly expanding and device software development, as well as companion apps, tend to be updated frequently [40]; thus we prioritized recent reviews, as older reviews may contain comments about outdated aspects of companion apps that might not apply anymore. A total of 585,078 Android and 27,784 iOS reviews survived these filtering steps.

3.2.7. Sample extraction

We extracted from $D_{Android}$ and D_{iOS} two samples of reviews of reasonable size to be analyzed manually. We opted for a sample size of 500 reviews. Such a sample size allows us to achieve a confidence level higher than 95% and a 5% confidence interval for both populations. From now on, we will refer to the extracted samples as $S_{Android}$ and S_{iOS}, respectively. In order to have a sample more representative of the complete population, we relied on a *stratified weighted sampling* procedure. In stratified sampling [30], the population is divided into subgroups, named *stratums*, by means of a criteria that partitions it. In our sampling procedure, we used the score associated with each review as partitioning criteria. Since scores are discrete values that range from a minimum of one to a maximum of five, we obtain a total of five stratums. Afterward, samples are extracted in the same proportion to the population from each stratum. The probability for each individual in a stratum of being selected is based on a given weight w. In our procedure, we defined the weight w_r for a review r related to an app a as $w_r = \frac{1}{1+n_a}$, where n_a is the number of reviews for app a in the stratum. In other words, the adopted weighting function assigns a greater weight to apps with fewer reviews in the stratum. We adopted the described sampling procedure for two reasons: (i) from a preliminary analysis of reviews in $D_{Android}$ and D_{iOS}, we observed that the distribution of the review scores is not uniform (as further discussed in Section 4.1), hence a stratified sampling procedure allowed us to preserve this distribution in the two samples $S_{Android}$ and S_{iOS}; (ii) the number of reviews for each app in $D_{Android}$ and D_{iOS} is highly unbalanced (as reported in Table 1), and therefore a selection based on the w_r weights was more likely to extract reviews belonging to a broader set of apps if compared with a fully random selection.

3.3. Data extraction and analysis

In this section, we describe how we extracted data from the obtained collection and analyzed the information obtained to answer our RQs.

3.3.1. Perceived quality over time (RQ1)

To provide an answer to RQ1, we performed an initial exploration of collected data by means of descriptive statistics and visualizations. We computed and analyzed time series for rates of different star ratings and sentiments over the years and across the two collected datasets $D_{Android}$ and D_{iOS}. The resulting time series are displayed in Figs. 3 through 6 and discussed in Section 4.1.

We conducted a statistical analysis of collected data to confirm the results of the initial exploration. For this reason, we considered, for both $D_{Android}$ and D_{iOS}, as *older reviews* those belonging to the most dated three years in our dataset (excluding years with ≤ 1000 collected reviews); whereas, we considered as *newer reviews* those belonging to the most recent three years. We tested for differences in the star rating and sentiment score between older and newer reviews, using the one-sided Mann–Whitney U test [41]. We selected this test as it does not assume the normality of the data being tested, an assumption that we know does not hold for star ratings since they can only assume discrete values from one to five. We formulated the null hypotheses H_0 = *"the distribution of star rating (sentiment score) for older and newer reviews are the same"* and the alternative hypotheses H_1 = *" the distribution of star rating (sentiment score) for older reviews is stochastically greater than newer reviews"*. When testing for differences in sentiment, we used the compound sentiment score computed by VADER as the sentiment value of each review.

3.3.2. Prominent issues (RQ2)

As for RQ2, the major difficulty resides in (i) uniquely identifying the subject of the review in order to understand if it is related to the companion app or the associated IoT device, and (ii) categorizing the main issues discussed by users with semantically significant labels to highlight the most critical aspects that require improvement. For that reason, in order to precisely answer RQ2, we resorted to manual analysis of the two extracted samples $S_{Android}$ and S_{iOS}, a procedure frequently used in software engineering studies [5,9,18,19]. To reduce bias, two experienced researchers took part in the analysis procedure, with each analyzing both complete samples independently. Each review was analyzed according to two different perspectives.

Content. Each review was classified according to its *content*, as:

- *App-related* (*A*): In this category we include reviews containing opinions about the app (e.g., *"Need some major update. Pretty empty app... Disappointed"*);
- *Device-related* (*D*): In this category we include reviews containing opinions about the IoT device (e.g., *"The camera is very slow to respond to motion [...]"*);
- *Unclear focus* (*U*): In this category we include reviews from which it is not possible to understand the focus (e.g., *"Wasting time with unnecessary functions"*).

Note that, since the *A* and *D* categories are intrinsically not mutually exclusive, having resorted to manual analysis allowed us to also clearly identify those reviews containing multiple opinions that refer to both the companion app and the IoT device (e.g., *"Serious frequent failures. Issues which cause the app and the Netgear access point to be frequently rebooted"*). Instead, reviews classified as having an unclear focus do not fall within either category.

This manual classification was conducted to: (i) verify that the collected user reviews indeed provide informative comments about companion apps and/or associated IoT devices, (ii) identify crosscutting categories, and (iii) build-up a ground truth for a future automation of the analysis process. Note that, understanding if an issue or malfunction depends on the device or app is relevant in order to identify the classes of problems to be addressed to improve the user perception.

With respect to this classification, we made use of the Krippendorff's Alpha coefficient [42] to measure the agreement between the two researchers before discussing and solving disagreement cases. We selected this measure for its ability to adjust itself to small sample sizes. We obtained an $\alpha = 0.81$ for $S_{Android}$ and an $\alpha = 0.73$ for S_{iOS}. Values of α over 0.6 are regarded as an adequate level of agreement, while values over 0.8 are considered as an indication of substantial agreement [43].

Concerns. Contextually with the described classification, the two reviewers also assigned descriptive labels to each surveyed review to summarize the *concerns* expressed in its content, following the guidelines of descriptive coding [44]. This second kind of analysis was conducted as an open labeling process, i.e., there was no predefined set of labels prior to starting the procedure. Rather, labels emerged naturally during the process, as more reviews were analyzed and recurring concerns identified. Potentially, multiple labels (if it expresses multiple concerns) or no labels (if uninformative) can be assigned to an individual review. After completing the analysis, the two researchers discussed together the assigned labels, aligned the used terminology (e.g., one researcher used a *"Bad UI"* label, while the other used a *"UI issue"* label to describe the same concept), and solved the cases for which there was a disagreement. Given the open nature of this labeling procedure and the need for a comparison between the two reviewers to realign the used terminology, it was not possible to use a metric to evaluate the agreement between the two, prior to discussing differences. Hence, to ensure the quality of the performed analysis, a third researcher was involved to break ties in those cases for which no agreement was reached during discussion. In most cases, the need for a tiebreaker was mostly due to the difficulties of labeling chaotic texts such as user reviews. Indeed, these often are short, omit important details, or lack correct sentence structure, which made it possible to label it differently according to the researcher's interpretation (e.g., *"Since the update I can no longer set a timer. It reverts to the next automatic change no matter what I do"*. was considered a *Broken update* by one researcher, while the other labeled it as a *Dark pattern* due to the removal of an existing feature). The intervention of the third researcher was required for 21 Android reviews and 14 iOS ones. The result of these activities was the definition of the taxonomy of smart-home mobile companion apps' user reviews in Fig. 2.

The content-related labels are described above; with respect to the concerns, a total of twenty-one descriptive labels have been used to describe recurring concerns found in the reviews. Furthermore, these have been organized in four macro-categories:

- *Functional-issues*: In this category we group together labels that describe issues of functional nature.
- *Non-functional issues*: In this category we group together labels that describe issues of non-functional nature.
- *Maintenance/evolution*: In this category we group together labels that describe issues related to the evolution and the maintenance of the application after its release.
- *User experiences*: In this category we group together labels used to describe negative user experiences of a less technical nature.

In Section 4.2, we present the analysis results and provide descriptions and examples for each label describing concerns.

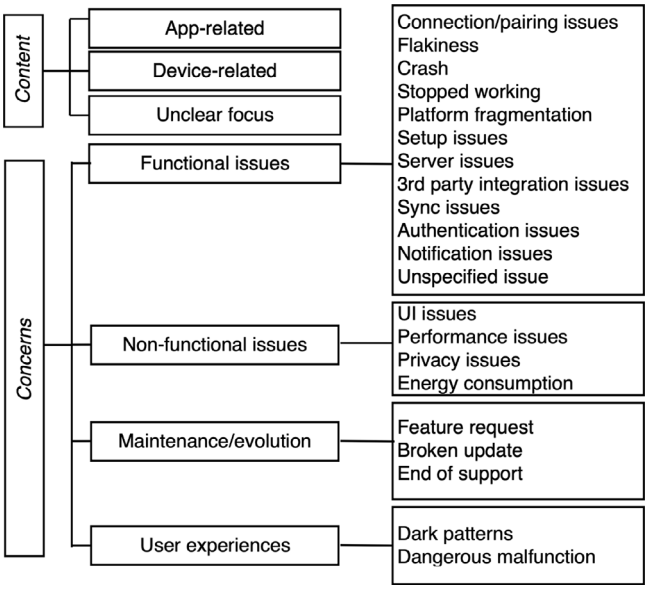


Fig. 2. Taxonomy of companion apps' user reviews.

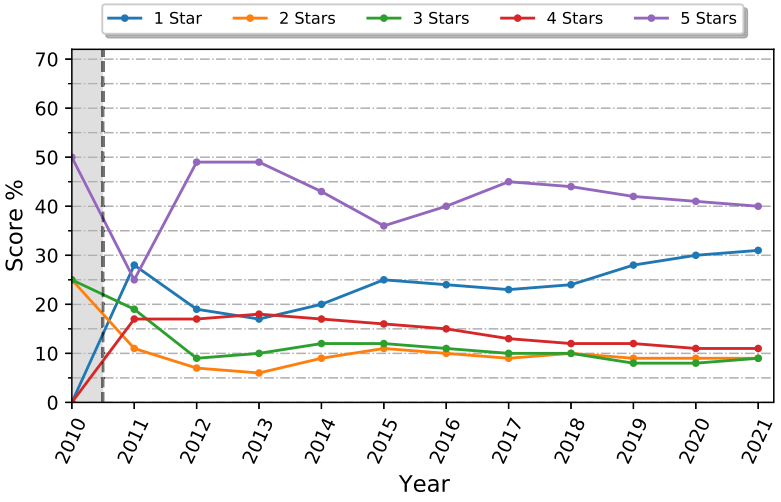


Fig. 3. Star rating of Android reviews over the years (years in grey have $\leq$ 1000 reviews).

4. Results

In the following, we present the results of the performed analysis to answer our research questions.

4.1. How has the perceived quality of smart-home IoT companion apps evolved over time?

Figs. 3 and 4 show the distribution of the star ratings over the years for $D_{Android}$ and D_{iOS}, respectively. For some years, only a limited amount ($\leq$ 1000) of scores have been collected, leading to greater variability of score ratios in these years. Thus, we will only partially consider these years in our analysis. We can observe, for both the Android and the iOS platform, a distribution of scores similar to the one of other consumer products [45], with a high number of score values at the extremes (i.e., one-star and five-star scores) and a considerably reduced frequency for value in the middle (i.e., from two-star to four-star scores).

Specifically, when focusing on Android, we observe that the ratio of five-star scores exhibits a decreasing trend: after reaching its maximum in the years 2012 and 2013 (49% for both years), it drops to its minimum in the year 2015 (36%)

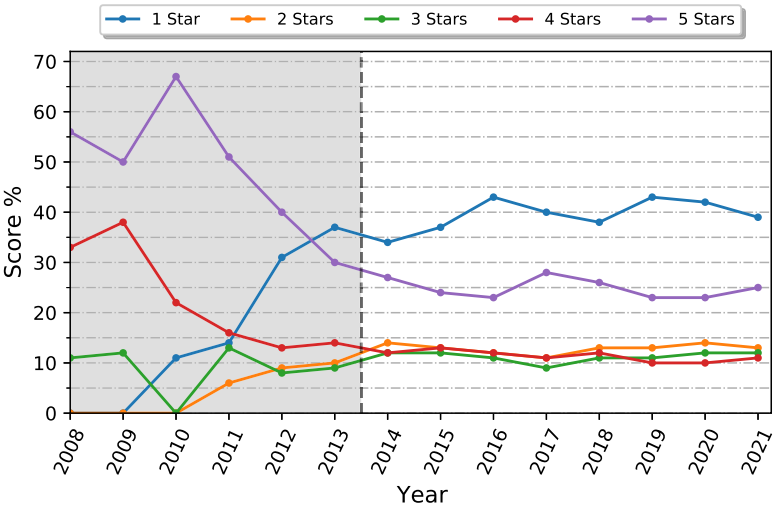


Fig. 4. Star rating of iOS reviews over the years (years in grey have $\leq$ 1000 reviews).

and then partially recovers, ending at 40% in 2021. Similarly, the ratio of four-star scores experiences a decreasing trend, ending at 11% in the year 2021 after an initial plateau for the years 2011 to 2013 (with values in the 13%–14% range). Regarding two and three-star scores, we observe a slightly higher ratio in the year 2011 (11% for two-star scores and 19% for three-star ones), followed by a decrease that leads to a minimum in the years 2012 and 2013 (7% for two-star scores and 9% for three-star ones), to then end at 9% for both scores in the year 2021. Finally, in contrast with the others, we observe a growing trend for one-star scores, which has an initial value of 28% in 2011, followed by a drop to its minimum in the year 2013 (17%), to then record a growing trend from the year 2014 onwards (20%), leading to the final and maximum value of 31% in 2021. Results of the one-sided Mann–Whitney U test allow us to reject the null hypothesis ($p-$ value < 0.01), thus confirming that, for Android applications, newer reviews (i.e., reviews in the 2019–2021 time frame) have a statistically significantly lower star rating than older reviews (2011–2013). The mean difference in the star ratings between the two groups amounts to -0.46.

Focusing on iOS apps, limited data is available until the year 2013. From the year 2014 onward, we can observe that the ratio of five-score ratings has a slightly decreasing tendency: initially at 27% in 2014, it experiences a drop in the following years, counterbalanced by an increase to 28% in 2017, and a final decrease until 2021, ending at 25%. An opposed increasing trend is observed for one-score reviews, that are at their minimum in the year 2014 (34%) and experience an increase over the years, registering their maximum in 2016 (43%) to then end at 39% for the year 2021. Ratios of other scores remain almost constant, starting in 2014 at 14% for two-star ratings, 12% for three-star ratings, and 12% for four star-ratings. In 2021, we observe ratios of 13%, 12%, and 11%, respectively. Similarly to Android applications, we can reject the null hypothesis of the one-sided Mann–Whitney U test ($p-$ value < 0.01), confirming that newer reviews (i.e., reviews in the range 2019–2021) have a statistically significantly lower star rating than older reviews (2014–2016). However, the mean difference in the star ratings between the two groups is more limited, amounting to -0.08.

The ratios of extracted sentiments over the years are provided in Fig. 5 for Android apps and in Fig. 6 for iOS ones. Regarding the former, we observe that the positive sentiment is the most common across all years. However, after starting at 60% for the year 2011 and reaching its peak in 2013 at 71%, it experiences a decreasing trend, recording its lowest value in 2021 at 59%. Specularly, the negative sentiment registers an increasing trend over the years, starting at 21%, registering its minimum in 2013 at 14% and then increasing until 2021 at 25%. Reviews with neutral sentiment constitute 19% of the total in the year 2011 and register a slight drop over the years, ending in 2021 at 16%. After testing the differences in sentiment scores between newer and older reviews, we can reject the null hypothesis ($p-$ value < 0.01). Hence, newer Android reviews have a statistically significantly lower sentiment score than older ones. The mean difference between the two groups is -0.12.

Similar trends can be observed for sentiments extracted from iOS reviews. The positive sentiment has a decreasing trend, constituting 66% of all extracted sentiments in 2014 (the first year for which sizeable data is available) and decreasing to 61% for 2021. In contrast, the negative sentiment records a ratio of 26% in 2014 and has a growing trend, ending at 32% in 2021. Noticeably, the ratio of reviews with neutral sentiment is considerably lower for iOS when compared to Android, oscillating in the 7%–9% range for all years for which sizeable data is available. This difference is likely attributable to the greater mean length of collected iOS reviews (73.85 mean words for iOS, as opposed to 21.96 for Android), which translates into a greater quantity of data on which the sentiment extraction is performed. Analogously to Android apps, after testing for differences in sentiment score between newer and older reviews, we can reject the null

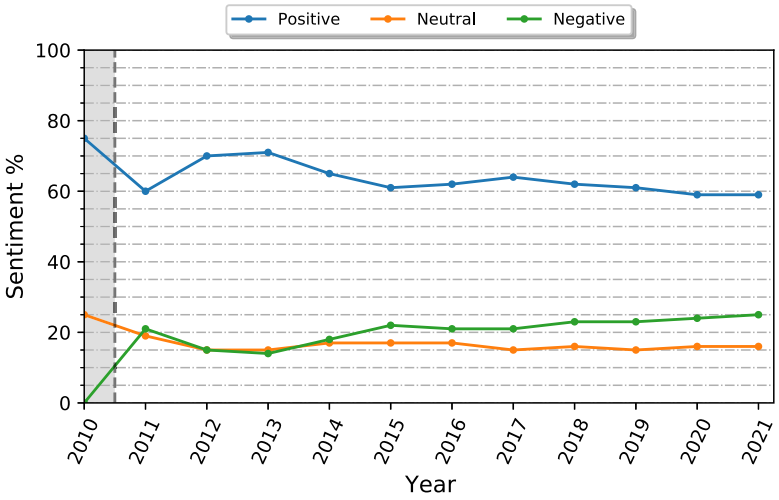


Fig. 5. Sentiment of Android reviews over the years (years in grey have $\leq$ 1000 reviews).

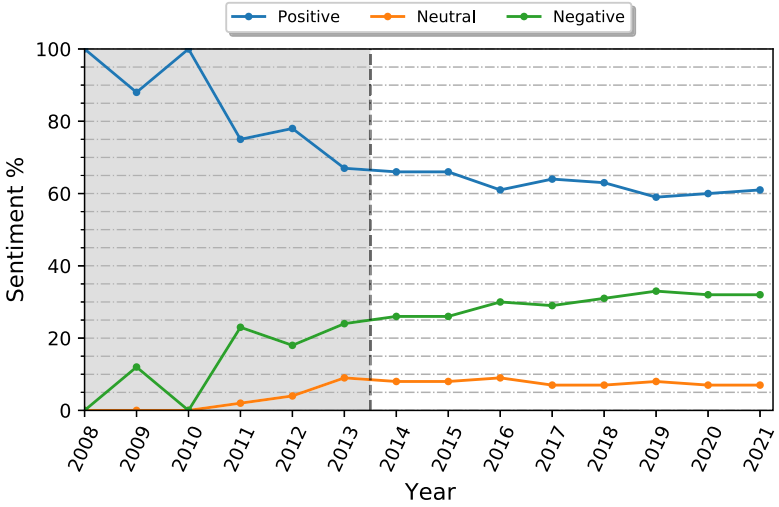


Fig. 6. Sentiment of iOS reviews over the years (years in grey have $\leq$ 1000 reviews).

hypothesis ($p-$ value < 0.01). Thus, for iOS apps, newer reviews have a statistically significantly lower sentiment score than older ones. The mean difference between the two groups is equal to -0.05.

4.2. Which are the prominent issues of smart-home IoT companion apps perceived by users?

In the following, we report on the results of the manual analysis performed on the two samples $S_{Android}$ and S_{iOS}. Within the former, 317 reviews were found to be app-related; whereas, 108 were found to be device related. Both totals include 47 reviews that have been classified as both app- and device-related. The remaining 122 reviews have been found to have an unclear focus. Regarding S_{iOS}, 348 reviews were classified as app-related and 185 as device-related, with 74 considered as both app- and device-related. The other 41 reviews in the sample have been found to have an unclear focus. Upon closer inspection, we notice that a lower amount of unclear reviews have been found in S_{iOS} compared to $S_{Android}$. Mostly, this discrepancy is due to the difference in length of reviews across the two platforms (with 73.85 mean words for iOS opposed to 21.96 for Android) that allows for a more easier understanding of their focus in the former.

In total, 423 labels have been assigned to 305 Android reviews (61% of all reviews in $S_{Android}$) and 593 labels have been assigned to 378 iOS ones (76% of S_{iOS}). Table 2 displays counts of occurrences for all the employed labels, divided by platform and with counts of app-related and device-related reviews. For each label, we also report its percentage in relation to the total amount of reviews in the sample.

Table 2
Breakdown of assigned labels.

		Android				iOS			
		A (%)	*D* (%)	*U* (%)	Total (%)	(*A* %)	*D* (%)	*U* (%)	Total (%)
Functional issues	Connection/pairing issues	37 (7.4%)	19 (3.8%)	8 (1.60%)	59 (11.8%)	63 (12.6%)	40 (8%)	6 (1.20)%	86 (17.2%)
	Flakiness	28 (5.6%)	9 (1.8%)	9 (1.8%)	41 (8.2%)	48 (9.6%)	22 (4.4%)	3 (0.6%)	58 (11.6%)
	Crash	22 (4.4%)	2 (0.4%)	1 (0.2%)	23 (4.6%)	30 (6%)	5 (1%)	1 (0.2%)	36 (7.2%)
	Stopped working	17 (3.4%)	8 (1.6%)	4 (0.8%)	24 (4.8%)	19 (3.8%)	14 (2.8%)	2 (0.4%)	32 (6.4%)
	Platform fragmentation	14 (2.8%)	3 (0.6%)	1 (0.20%)	18 (3.6%)	24 (4.8%)	1 (0.20%)	0 (0%)	24 (4.8%)
	Setup issues	14 (2.8%)	6 (1.2%)	5 (1%)	24 (4.8%)	12 (2.4%)	11 (2.2%)	0 (0%)	21 (4.2%)
	Server issues	4 (0.8%)	2 (0.4%)	1 (0.20%)	6 (1.2%)	9 (1.8%)	4 (0.8%)	1 (0.20%)	12 (2.4%)
	3rd party integration issues	8 (1.6%)	0 (0%)	4 (0.8%)	12 (2.4%)	7 (1.4%)	3 (0.6%)	1 (0.2%)	10 (2%)
	Sync issues	4 (0.8%)	3 (0.6%)	1 (0.2%)	6 (1.2%)	8 (1.6%)	1 (0.2%)	1 (0.2%)	10 (2%)
	Authentication issues	13 (2.6%)	2 (0.4%)	0 (0%)	13 (2.6%)	7 (1.4%)	1 (0.2%)	2 (0.4%)	10 (2%)
	Notification issues	9 (1.8%)	2 (0.4%)	0 (0%)	10 (2%)	6 (1.2%)	3 (0.6%)	0 (0%)	7 (1.4%)
	Unspecified issue	16 (3.2%)	3 (0.6%)	6 (1.2%)	22 (4.4%)	20 (4%)	9 (1.8%)	3 (0.6%)	29 (5.8%)
Non-functional issues	UI issues	25 (5%)	6 (1.2%)	4 (0.8%)	31 (6.2%)	54 (10.8%)	14 (2.8%)	0 (0%)	54 (10.8%)
	Performance issues	12 (2.4%)	9 (1.8%)	2 (0.4%)	17 (3.4%)	15 (3%)	7 (1.4%)	2 (0.4%)	18 (3.6%)
	Privacy issues	5 (1%)	1 (0.2%)	1 (0.2%)	6 (1.2%)	10 (2%)	5 (1%)	0 (0%)	11 (2.2%)
	Energy consumption	1 (0.2%)	0 (0%)	0 (0%)	1 (0.2%)	4 (0.8%)	2 (0.4%)	0 (0%)	4 (0.8%)
Maintenance/evolution	Feature request	39 (7.8%)	5 (1%)	9 (1.80%)	53 (10.6%)	57 (11.4%)	20 (4%)	3 (0.6%)	68 (13.6%)
	Broken update	25 (5%)	5 (1%)	3 (0.6%)	31 (6.2%)	44 (8.8%)	5 (1%)	2 (0.4%)	48 (9.5%)
	End of support	10 (2%)	2 (0.4%)	5 (1%)	15 (3%)	17 (3.4%)	12 (2.4%)	0 (0%)	25 (5%)
User experiences	Dark patterns	8 (1.6%)	1 (0.2%)	1 (0.2%)	10 (2%)	19 (3.8%)	11 (2.2%)	1 (0.2%)	26 (5.2%)
	Dangerous malfunction	2 (0.4%)	3 (0.6%)	0 (0%)	4 (0.8%)	2 (0.4%)	1 (0.2%)	0 (0%)	3 (0.6%)

A = App-related, *D* = Device-related, *U* = Unclear focus.

4.2.1. Functional issues

These issues constitute the bulk of identified labels, with a 258 functional issues identified for Android apps, and 335 for iOS ones.

Connection/pairing issues — This label describes issues in establishing a connection between the companion app and the IoT device, that render impossible its remote use. This is the most frequent functional issue, found in 59 Android reviews and 86 iOS ones, which translate to 11.8% and 17.2% of all reviews in the respective samples. An example is:

> *"Awful.. the app is constantly loosing connection with the bulb meaning I have to switch off at the wall and back on again to reset the connection. It's there one minute and gone the next! Very frustrating!"*

Flakiness — This label is assigned to reviews in which users describe an unreliable behavior of the companion app or its associated device, characterized by alternating periods of regular operation and periods of non-operation with no apparent explanation. Problems of this kind have a strong negative impact on the user experience and have been identified in 41 Android reviews and 58 iOS ones. An example is the following:

> *"App regularly crashes, randomly my devices will turn off, reset and just not record, even if it is scheduled to. [...] Cant view or upload clips or pics that are saved internally. In short, the thing does what it wants, when it wants, and never is reliable when I need it and has been that way since i got it".*

Crash — This label is assigned to reviews in which users report experiencing an abrupt termination app due to a software bug. We observed reports of this behavior in 23 Android and 36 iOS reviews. An example is the review below:

> *"Continuously crashing upon opening or when attempting to edit a zone. Total frustration. Application technical support has been useless".*

Stopped working — This label groups those reviews in which users describe that the device or companion app associated with it has unexpectedly stopped working after a limited period of time. Overall, 24 Android and 32 iOS reviews have been identified that belong to this category. An example is provided below:

> *"This app is horrible. I bought two smart plugs and they connected to my phone for a few days and then just unexpectedly stopped working. [...]"*

Platform fragmentation — This label is used to denote those reviews in which users report poor compatibility of the companion app on certain mobile devices [46]. This kind of issues have been found in 18 Android and 24 iOS reviews. An example of reviews of this kind is the following:

> *"App won't connect to my LG Soundbar anymore since upgrading to iOS 14. Come in LG, get your act together!!!"*

Setup issues — This label groups those reviews in which users report problems experienced during the installation or the first usage of the device. Overall we have identified 24 Android reviews and 21 iOS ones that belong to this category. An example is the following:

> *"Out of home controller has delays. Device setup is a nightmare"*

Server issues — This label is used for those reviews which describe problems attributable to the malfunctioning of the remote servers to which the computation is offloaded by the IoT device or the companion app. Overall, 6 Android and 12 iOS reviews have been identified that report problems of this type. Below is an example:

> *"Their product does not even work, the vents do not automatically close by themselves. They claim it's possibly a server issue but it's been a while now. [...]"*

Third-party integration issues — This label is used to highlight those reviews that describe problems in the interaction between the IoT product and third-party services. 12 Android and 10 iOS instances have been identified for this label. An example is the following:

> *"[...] My big issue is that this thermostat claimed to be Alexa capable yet I can never get it to work. Alexa never recognizes the thermostat, which is terribly frustrating. I have also tried connecting the thermostat to Apple HomeKit to see if it worked better than Alexa but I can't even get HoneKit to recognize the thermostat. [...]"*

Sync issues — This label is used for those reviews describing data synchronization problems between the companion app and the IoT device, resulting in a misalignment between the two. Overall, 6 Android and 10 iOS instances of this type of issue were identified. A clarifying example is below:

> *"Every time I turn off devices all the configuration is gone. I have to setup everything from beginning. I have updated my firmware. App hangs very often".*

Authentication issues — This label groups those reviews in which users report problems related to user authentication, which undermines the user experience. Overall, 13 Android and 10 iOS instances of this kind have been identified. An example is the following:

> *"My username and password are correct as I can log in via the website, but the app says the password is wrong and won't let me reset it because it says the email is wrong. Then it won't let me sign up with my email because it says something is wrong! [...]"*

Notification issues — This label groups those reviews in which users report problems related to push notifications produced by the companion app. We can identify two main cases within it. In the first case users report receiving an excessive number of notifications, hence perceiving them as spam; In the second case instead users report not receiving any notification or receiving them with an excessive delay, hence missing important events. Problems of this kind have been found in 10 Android and 7 iOS reviews. An example is provided below:

> *"This was a great app but recently started sending notifications for motion sensing which I've never enabled. Now getting flooded with useless push notifications".*

Unspecified issue — This label is used to denote those reviews in which users express complaints about the application but do not provide enough details to fully understand the nature of the problem. Overall, 22 Android and 29 iOS reviews of this type have been identified. An example is the following:

> *"Kind of like a program written by a high school student. I'm taking back the device and finding something better, with a better app".*

4.2.2. Non-functional issues

These have been found in a minority of analyzed reviews, composed of 55 Android reviews and 87 iOS ones.

UI issues — This label identifies reviews that complain about poor User Interfaces (UI), considering them too complicated or deemed to be deficient, thus rendering the application and associated service difficult to use. A total of 31 Android and 54 iOS reviews that describe complaints of this kind have been identified, corresponding to 6.2% and 10.8% of reviews in their respective samples. An example is the following:

> *"This app is horrible. Period. I had been fussing with it for months, trying to get Logitech support to help me set it up. The app is completely incomprehensible and non-user friendly. Furthermore, Logitech's technology does not work at all on any network!"*

Performance issues — This label is used to identify reviews in which users describe experiencing issues related to app performance, such as response time, and resource consumption [13]. Issues of this kind have been found in 17 Android and 18 iOS reviews. An example is the following:

> *"The idea and design maybe good but not as good as it should be. Very slow and delayed video even within Wi-Fi network, forget in remote mode".*

Privacy issues — This label is used to evidence reviews in which users express a concern about their privacy, either described as being requested an excessive amount of personal data to use the service or by being unable to understand the reasons behind the collection of some sensitive data. In total, 6 Android and 11 iOS apps have been marked with this label. An example of a review of this kind is the following:

> *"Why should I always need location on to use this bulb? I understand using location service ONCE while pairing the device but don't understand the need to keep location on always".*

Energy consumption — This label is used for reviews in which complaints about an excessive energy consumption are found, resulting in a reduced battery life of the mobile or IoT device. This label was identified in a limited number of reviews, comprised of 1 Android and 4 iOS instances.

> *"This app is now using 25% of my battery. This must change since im already invested into abode".*

4.2.3. Maintenance/evolution issues

These amount to the second most common category of issues, with 99 identified for Android apps and 141 for iOS ones.

Feature requests — This label is assigned to reviews that contain suggestions for new features or improvements of existing ones. Indeed, it is well known that user reviews can be an useful source of information to derive requirements of future app releases [7,8,17]. Noticeably, in the case of companion apps, suggestions may not be limited to the application alone but may extend to the associated device. A total of 68 iOS and 53 Android reviews have been identified belonging to this category. An example of a review that includes a request of improvement is the following:

> *"I've love my system and the app works great, but I wish the schedule feature allowed on/off settings. [...]"*

Broken update — This label identifies reviews that report the insurgence of an issue after updating the companion app or the device firmware. Issues of this kind have been found in 31 Android and 48 iOS reviews which correspond to 6.2% and 9.6% of all reviews in $S_{Android}$ and S_{iOS}, respectively. An example is given below:

> *"I can not turn on/off motion detector remotely with latest update. It's always been a pain but was finally working ok, now it won't connect. [...]"*

End of support — This label is used to describe user reviews that report that the companion app and/or the IoT device no longer work, due to its support having been discontinued by the proprietary company. A total of 15 Android and 25 iOS reviews have been assigned this label. A clarifying example is the following:

> *"They SkyBell stopped working because they stopped supporting the technology to make you buy a new door bell. I only had it for a couple of years. That's ridiculous and I won't get another one"*

4.2.4. User experiences

This category groups together two labels used to describe negative user experiences of a less technical nature, found in 14 Android reviews and 29 iOS ones. We consider these user-reported experiences relevant to our analysis due to their potential socio-technical impact (discussed in Section 5).

Dark pattern — The term *"Dark pattern"* identifies instances where designers use their knowledge of human behavior (e.g., psychology) and the desires of end users to implement deceptive functionality that is not in the user's best interest [47]. Initially defined in the context of screen-based interactions [48], the concept of dark pattern is broader and can be expanded to all interactions that affect the user experience [47,49]. In the course of our analysis, we have identified 9 Android and 25 iOS reviews that describe dark patterns. Analyzing them more in-depth, we have identified two types of dark patterns that are more frequent. The first, found in 5 Android and 12 iOS reviews, encompasses those cases in which to push users towards the sign-up of a monthly subscription, existing system functionalities are removed and locked behind a paywall. This behavior is known as *"forced action"* accordingly to the categorization of dark patterns provided by Gray et al. [47]. An example is the following review:

> *"Used to be that you can view all locally recorded events from your server. Now you have to subscribe to their cloud service! Feature taken away just to force subscription. [...]"*

The second dark pattern frequently identified, found in 2 Android and 5 iOS reviews, concerns those cases in which some relevant information is omitted or purposely delayed to the user. This dark pattern is known as *"sneaking"* in the categorization provided by Gray [47]. An example is given below:

> *"Charging people 3$ monthly for the implementation of geo fencing and window detection is too much and should be stated much clearer on the product when purchasing. [...] "*

The remaining cases are varied and include, among others, the inability to disable certain functionalities (e.g., *"Developers, create a button to turn off "by the way" suggestions [...]. I just want a simple answer from the device so I can get on with life".*), flooding the user with messages (e.g., *"Won't stop spamming me to get a review"*), and making it difficult to cancel an ongoing subscription.

Dangerous malfunction — This label has been assigned to those reviews that describe situations in which users have perceived a potential danger due to a malfunction of the IoT device or of the companion app. In total, we identified 4 Android and 3 iOS reviews marked with this label. An example is the following, in which a user describes the malfunctioning of a smart thermostat:

> *"Still one star... Thermostat starts overheating since a while now, passing the set temperature without turning off. Which is very dangerous seen I have one in the newborn baby room. [...]"*

5. Discussion

In the following, we discuss the results provided in the previous section to call attention to their impact.

Perceived quality has not improved over the years. The results of RQ1 highlight that the perceived quality of smart-home companion apps has not improved over the years. Statistical analysis confirms a decrease in both star rating and sentiment score in more recent years. Indeed, the ratios of positive star rating scores and positive sentiment do not record an increasing trend over the observed time span, which covers a period of over ten years. This highlights that the companion apps and related IoT devices have not matured over the years, and are still affected by a variety of problems of different nature, described in the results of the RQ2. User complaints have been found in 61% of reviews in the analyzed Android sample and 76% of reviews in the iOS one. These numbers, combined with the issues identified in RQ2, show that smart-home companion apps software is flaky, unoptimized, and difficult to use, thus failing to meet end-users' expectations. Although more research is required to fully comprehend the reasons behind this lack of evolution, we hypothesize that this is due to the fact that developing quality software for IoT products is inherently difficult, as it relies on the interplay of heterogeneous devices (i.e., cloud, low-powered and mobile devices) and it requires ensuring a wide set of non-functional qualities (e.g., usability, reliability, privacy).

Most frequent issues We collected a large number of functional issues. Among them, the most argued are *Connection/pairing issues* (19.34% of labeled Android reviews and 22.75% of iOS ones), *Flakiness* (13.44% Android and 15.34% iOS), *Crash* (7.54% Android and 9.52% iOS), *Stopped working* (7.87% Android and 8.47% iOS), and finally the generic *Unspecified bug* (7.21% Android and 7.67% iOS). Whereas, among the non-functional issues, user interfaces *UI issues* stand out with 10.16% of labeled Android reviews and the 14.29% of iOS ones. Such results underline that having a usable, functional and efficient companion app is essential for the success of the device. Other common discussed features are related to maintenance and evolution, that are *Feature request* (17.38% Android and 17.99% iOS) and *Broken update* (10.16% Android and 12.7% iOS). The former highlights that user reviews of companion apps are a valuable source of information to derive requirements for future releases.

Privacy is rarely considered Focusing on the results of our qualitative analysis (described in Section 4.2), we can observe that a limited amount of reviews express privacy-related concerns, identified in only 6 Android and 11 iOS reviews. Although this is in agreement with previous studies, that found that only a limited fraction of mobile app reviews discuss privacy concerns [50–52], this absence is particularly relevant in the context of IoT devices, which possess a wide range of always-on sensors (e.g., microphone, camera, GPS), potentially usable to acquire a variety of sensitive information. A possible explanation for this phenomenon is that the more privacy-concerned individuals refrain from adopting smart-home IoT devices, driven by their privacy concerns. However, this also highlights that most consumers do not pay special attention to privacy-related aspects of IoT devices, or are willing to accept compromises in this regard, trading privacy in exchange for more advanced features. Consequently, we reason that, in the IoT domain, to achieve effective privacy cannot protection, it cannot be delegated to end-users but has to be enforced by legislators and platform administrators through specific rules and legislations.

Socio-technical challenges Some identified issues go beyond the technical sphere and can have a broader impact on society. In particular, the reviews tagged with the *Dangerous malfunction* label evidence that, since IoT devices are

equipped not only with sensors but also with actuators, they can potentially become a danger in the eventuality of a software malfunction. Although software whose malfunction can have critical consequences is not new [53], it must be taken into account that in the future IoT devices are expected to be extremely widespread and pervasive. Therefore, the challenge of ensuring the safety of these devices on a large scale arises.

A second noteworthy problem is the presence of dark patterns, traditionally relegated to digital interactions and on the Web [48]. Presence of dark patterns in commercial IoT devices highlights their passage from the purely digital world to the physical world, where they could potentially creep into interactions with appliances and devices of daily usage. However, different from the digital world, installation and configuration of IoT products is more expensive and time-consuming. Therefore the balance of power is further skewed towards product developers that could more easily coerce users into decisions not in their complete interest.

Future research challenges Based on our findings and considerations, we highlight the following research challenges:

- Designing new techniques, tools, and methodologies to assist developers in the construction and maintenance of IoT products. The results of our study evidence that a considerable amount of functional issues derive from the difficult interplay of low-powered, cloud, and mobile devices. Indeed, testing IoT systems is a challenging task, due to the cross-domain particularities of these systems, the considerable number of involved devices, the unreliable connectivity, and device and protocols heterogeneity [54–56]. While dedicated solutions do exist [57,58], these fail to fully address existing challenges, and further effort is required towards the development of testing solutions, automation procedures, and continuous integration features specifically tailored for IoT systems [54]. Moreover, ensuring adequate quality levels is difficult, as a multitude of aspects needs to be considered [56,59], including availability, performance, privacy, security, and energy consumption. Indeed, in the results of our qualitative analysis (refer to Section 4.2) we identified user concerns related to these aspects in user reviews. Further work is needed for defining frameworks to measure and ensure the necessary quality levels of selected aspects of IoT systems [56]. To this end, akin to regular mobile apps [15], monitoring user reviews of companion apps can provide guidance on aspects that require improvement. Furthermore, in the smart-home context, the development of integrations among different IoT devices also poses a significant challenge [60], as developers need to properly handle the heterogeneity of different devices, possible situations, and errors. Hence, there is a need for standards and tools that facilitate the integration of such heterogeneous software and hardware components, frequently developed and commercialized by different vendors.
- Improving procedures for the release of software updates. In the smart-home domain, software updates are deployed over the air for both the IoT device firmware and the associated companion app, to add new features, resolve software bugs, and address security vulnerabilities [61]. In our qualitative analysis, we found that broken updates are a significant source of complaints for users of companion apps. While in other domains it is possible to quickly distribute urgent bug fixes to address issues discovered after the release of a update [62,63], this might not always be possible for IoT devices, due to limited processing power and intermittent connectivity [61,64]. Hence, it is necessary to design procedures for the safe roll-out of updates, e.g., adapting change impact analysis techniques [65] to the IoT domain and devising robust offline rollback procedures [66]. In addition to reducing issues caused by updates, this would partially reduce the effort required by manufacturers to support devices, thus meeting the expectations of users who demand for IoT devices a longer lifetime than what is currently guaranteed by manufacturers. Additionally, it is necessary to develop a better understanding of how users perceive software updates: although it is known that the updates release strategy can affect the success of a mobile application [67], it is unclear if a similar effect also exists for smart-home devices.
- In relation to privacy, due to the novelty of the technology, users' concerns are not yet fully understood [28,68]. While not focusing on privacy alone, our study shows that user reviews of companion apps can represent a useful source of information to investigate user concerns *in-the-wild*. Moreover, there is a need for more effective ways to convey to users the privacy risks that threaten them. While attempts do exist [69], the effectiveness of these visualizations in nudging actual users' behavior needs to be investigated [70]. Furthermore, the solutions that have been proposed aim to inform users at the time of device purchase, but privacy and security threats can arise at a later time during the device lifetime, due to misconfigurations [71] or software updates [72]. Hence, devising solutions for informing and assisting users during these later phases is an open research challenge. Additionally, several studies have investigated solutions to introduce access control capabilities into smart-home IoT systems [73–75]. However, due to the heterogeneity of these devices, there is no one-size-fits-all solution [75] and mechanisms to elicit privacy preferences while preserving an acceptable user experience are required [74]. Finally, it is necessary to investigate which are the most common violations, in order to gather evidence for regulators and platform maintainers to introduce new privacy-preserving rules.
- It is necessary to achieve a deeper understanding of dark patterns and strategies used to coerce users into decisions, not in their complete interest. Researchers have documented the presence of dark patterns in interactions with smart devices in shared public spaces [76], and our work provides preliminary evidence of their presence in interactions with smart appliances in private spaces. While efforts have been conducted to catalog and study the prevalence of dark patterns in mobile applications [77], it is unclear how frequent and impactful these are in companion apps. Furthermore, in the context of online interactions, users have been found to be generally aware of the influence of dark patterns on their behavior but, nonetheless, they are unable to oppose such influence [78,79]. Hence, given the pervasiveness of IoT systems, these threats pose an increased risk in this new domain. For this reason, it is important to build a deeper understanding of dark patterns in IoT, which will help in designing new rules and countermeasures to restore the balance of power between users and manufacturers.

6. Threats to validity

In the following, we discuss the threats to the validity of our study according to the Cook and Campbell categorization [80].

Internal validity refers to the causality relationship between treatment and outcome. In our study, we relied on an automated tool to identify the sentiment of user reviews. Therefore, we rely on the tool correctness, and thus, its potential issues could affect our study results. To mitigate this risk, we purposely selected a tool frequently employed in opinion mining studies [37] and specialized to deal with short texts. Moreover, a manual procedure was employed to identify user concerns. As with all kinds of manual processes, mistakes might have occurred. To mitigate this threat, two different researchers performed this task independently and a third one was involved to break ties. When applicable, their agreement level was measured with the Krippendorff alpha [43] and resulted satisfactory. Quantitative data collected corroborates the results of the manual analysis.

External validity deals with the generalizability of obtained results. To ensure that our subjects are representative of the population of companion apps, we verified their number of installations and reviews, provided in Table 1, before proceeding with our analysis. In addition, we collected and analyzed samples of both Android and iOS companion apps to ensure that concerns of users of both platforms are considered in our study. Finally, we adopted a stratified sampling procedure to ensure that analyzed reviews are representative of a wider range of user judgments.

Construct validity deals with the relation between theory and observation. In our study, we analyze user reviews of companion apps, to identify recurrent issues from the end-user perspective. A wide variety of devices offers an associated companion apps and each might be subject to different issues. When dealing with subtle issues, only a minority of users show sufficient awareness. Hence, the proposed approach might not discover the more subtle issues. We mitigate these threats by analyzing two statistically relevant sets of reviews, that allow us to achieve a confidence level higher than 95% and a 5% confidence interval.

Conclusion validity deals with issues that affect the ability to draw the correct conclusions from the outcome of experiments. To mitigate this threat, while answering RQ1, we complemented our exploratory analysis with the usage of statistical tests to prove our assumptions and test our hypotheses and therefore limit the room for error when interpreting the experiment results. In addition, in our study, we conducted a qualitative analysis of two reviews samples to build an understanding of perceived issues and provide confidence in our results. Furthermore, in this study, we assumed that user reviews are a reliable source for inferring user concerns. However, there may be other factors that potentially may affect users' judgment. Numerous examples of purposeful reviews have been reported in the paper to highlight the usefulness of classified concerns. The full set of classified user reviews is publicly available in the replication package of this study.

Reliability validity concern the extent to which the obtained findings can be reproduced. To mitigate this threat We are making available the collected datasets plus mining and analysis scripts, other than providing full details about the data extraction and analysis procedures.

7. Conclusion and future work

We conducted an empirical study of 1,347,799 Android and 48,498 iOS user reviews, to investigate the perceived quality and prominent issues of IoT mobile companion apps. Combining qualitative and quantitative methods, we uncovered that users' judgment has not improved over the years, due to a variety of functional and non-functional issues, such as difficulties in paring with the device, software flakiness, poor user interfaces, and issues of a socio-technical impact. Based on our findings, we identified open research directions to address aspects that require improvement in order to meet user expectations.

The manual analysis, presented in this work, was performed with the aim to build up a ground truth for future automation of the analysis process. As future work, we plan to extend this work through an automated and improved process. App review analyzes can be automated using different text mining techniques, mainly based on Machine Learning (ML) and Natural Language Processing (NLP) [38].

With the aim to prioritize the user reviews, we will consider to use a tool for automatically filtering and ranking informative reviews [39]. In order to answer RQ1, we already employed automated tools to compute and analyze sentiments. For RQ2, we performed a manual analysis with the aim of (i) uniquely identifying the subject of the review, and (ii) categorizing the main issues discussed by users. Such steps can be automated by employing techniques for clustering (e.g., to group reviews discussing the same topics) and classification (e.g., to categorize user feedback based on user intention) [38]. Also, we plan to extend the sentiment analysis to identify feature-specific sentiment. Whereas, recommendation tools can be adopted to assign priorities to reviews reporting bugs [81] and information extraction techniques to identify features [38].

Moreover, some existing approaches can be considered to extend this work. Among them, [82] proposes an approach for capturing user needs useful for developers performing maintenance and evolution tasks. In particular, the approach automatically (i) extracts the topics treated in reviews, (ii) classifies the intention of the writers, to suggest the specific kinds of maintenance tasks developers have to accomplish, and (iii) groups together sentences covering the same topic.

Finally, we are interested in studying how the (planned) automated analysis can be turned into a continuous monitoring effort. Detecting users' significant intentions (e.g., new features wanted) timely and precisely is crucial, also users' sentiment and preferences often change over time due to either internal factors (e.g., new bugs) or external factors (e.g., new competitors). In this respect, the temporal correlation between user review results can be analyzed; for instance, in [83], NLP techniques are applied to obtain sentence-level sentiment scores and fine-grained user preference features from app reviews in different time slices. Moreover, incremental learning techniques can be applied, so that updated data can be continuously used to extend the existing model's knowledge [84]. Tracking how user reviews evolve over time would provide a better knowledge base to improve apps accordingly and continuously.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data is available at https://bit.ly/companionApps.

Acknowledgments

This work was partially supported by: the Italian Government under CIPE resolution n. 135 (December 21, 2012), project INnovating CIty Planning through Information and Communication Technologies (INCIPICT); the Italian MIUR SISMA national research project (PRIN 2017, Contract 201752ENYB); the Italian PNRR MUR Centro Nazionale HPC, Big Data e Quantum Computing, Spoke9 - Digital Society & Smart Cities; and the AIDOaRt project grant from the ECSEL Joint Undertaking (JU) (grant n. 101007350).

References

[1] Azana Hafizah Mohd Aman, Elaheh Yadegaridehkordi, Zainab Senan Attarbashi, Rosilah Hassan, Yong-Jin Park, A survey on trend and classification of Internet of Things reviews, IEEE Access 8 (2020) 111763–111782, http://dx.doi.org/10.1109/ACCESS.2020.3002932.
[2] Xueqiang Wang, Yuqiong Sun, Susanta Nanda, XiaoFeng Wang, Looking from the mirror: Evaluating {*IoT*} device security through mobile companion apps, in: 28th USENIX Security Symposium (USENIX Security 19), 2019, pp. 1151–1167.
[3] Abhinav Mohanty, Meera Sridhar, HybriDiagnostics: Evaluating security issues in hybrid SmartHome companion apps, in: 2021 IEEE Security and Privacy Workshops, SPW, IEEE, 2021, pp. 228–234.
[4] Necmiye Genc-Nayebi, Alain Abran, A systematic literature review: Opinion mining studies from mobile app store user reviews, J. Syst. Softw. 125 (2017) 207–219.
[5] Jacek Dąbrowski, Emmanuel Letier, Anna Perini, Angelo Susi, Analysing app reviews for software engineering: a systematic literature review, Empir. Softw. Eng. 27 (2) (2022) 1–63.
[6] Emitza Guzman, Walid Maalej, How do users like this feature? a fine grained sentiment analysis of app reviews, in: 2014 IEEE 22nd International Requirements Engineering Conference, RE, Ieee, 2014, pp. 153–162.
[7] Walid Maalej, Hadeer Nabil, Bug report, feature request, or simply praise? on automatically classifying app reviews, in: 2015 IEEE 23rd International Requirements Engineering Conference, RE, IEEE, 2015, pp. 116–125.
[8] Walid Maalej, Zijad Kurtanović, Hadeer Nabil, Christoph Stanik, On the automatic classification of app reviews, Requir. Eng. 21 (3) (2016) 311–331.
[9] Jacek Dąbrowski, Emmanuel Letier, Anna Perini, Angelo Susi, Mining user opinions to support requirement engineering: an empirical study, in: International Conference on Advanced Information Systems Engineering, Springer, 2020, pp. 401–416.
[10] Cuiyun Gao, Wujie Zheng, Yuetang Deng, David Lo, Jichuan Zeng, Michael R. Lyu, Irwin King, Emerging app issue identification from user feedback: Experience on wechat, in: 2019 IEEE/ACM 41st International Conference on Software Engineering: Software Engineering in Practice (ICSE-SEIP), IEEE, 2019, pp. 279–288.
[11] Hui Yang, Peng Liang, Identification and classification of requirements from app user reviews, in: SEKE, 2015, pp. 7–12.
[12] Mengmeng Lu, Peng Liang, Automatic classification of non-functional requirements from augmented app user reviews, in: Proceedings of the 21st International Conference on Evaluation and Assessment in Software Engineering, 2017, pp. 344–353.
[13] Nishant Jha, Anas Mahmoud, Mining non-functional requirements from app store reviews, Empir. Softw. Eng. 24 (6) (2019) 3659–3695.
[14] Emitza Guzman, Muhammad El-Haliby, Bernd Bruegge, Ensemble methods for app review classification: An approach for software evolution (n), in: 2015 30th IEEE/ACM International Conference on Automated Software Engineering, ASE, IEEE, 2015, pp. 771–776.
[15] Sebastiano Panichella, Andrea Di Sorbo, Emitza Guzman, Corrado A. Visaggio, Gerardo Canfora, Harald C. Gall, How can i improve my app? classifying user reviews for software maintenance and evolution, in: 2015 IEEE International Conference on Software Maintenance and Evolution, ICSME, IEEE, 2015, pp. 281–290.
[16] Lorenzo Villarroel, Gabriele Bavota, Barbara Russo, Rocco Oliveto, Massimiliano Di Penta, Release planning of mobile apps based on user reviews, in: 2016 IEEE/ACM 38th International Conference on Software Engineering, ICSE, IEEE, 2016, pp. 14–24.
[17] Andrea Di Sorbo, Sebastiano Panichella, Carol V. Alexandru, Junji Shimagaki, Corrado A. Visaggio, Gerardo Canfora, Harald C. Gall, What would users change in my app? summarizing app reviews for recommending software changes, in: Proceedings of the 2016 24th ACM SIGSOFT International Symposium on Foundations of Software Engineering, 2016a, pp. 499–510.
[18] Grant Williams, Anas Mahmoud, Modeling user concerns in the app store: A case study on the rise and fall of yik yak, in: 2018 IEEE 26th International Requirements Engineering Conference, RE, vol. 6, IEEE, 2018, pp. 4–75.
[19] Artemij Voskobojnikov, Oliver Wiese, Masoud Mehrabi Koushki, Volker Roth, Konstantin Beznosov, The u in crypto stands for usable: An empirical study of user experience with mobile cryptocurrency wallets, in: Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems, 2021, pp. 1–14.

[20] Suhaib Mujahid, Giancarlo Sierra, Rabe Abdalkareem, Emad Shihab, Weiyi Shang, Examining user complaints of wearable apps: A case study on android wear, in: 2017 IEEE/ACM 4th International Conference on Mobile Software Engineering and Systems (MOBILESoft), IEEE, 2017, pp. 96–99.
[21] Suhaib Mujahid, Giancarlo Sierra, Rabe Abdalkareem, Emad Shihab, Weiyi Shang, An empirical study of android wear user complaints, Empir. Softw. Eng. 23 (6) (2018) 3476–3502.
[22] Vahid Garousi, David Cutting, Michael Felderer, What do users think of COVID-19 contact-tracing apps? An analysis of eight European apps, IEEE Softw. (2021).
[23] Xabier Larrucea, Annie Combelles, John Favaro, Kunal Taneja, Software engineering for the internet of things, IEEE Softw. 34 (1) (2017) 24–28.
[24] Amir Makhshari, Ali Mesbah, IoT bugs and development challenges, in: 2021 IEEE/ACM 43rd International Conference on Software Engineering, ICSE, IEEE, 2021, pp. 460–472.
[25] Fulvio Corno, Luigi De Russis, Juan Pablo Sáenz, On the challenges novice programmers experience in developing IoT systems: A survey, J. Syst. Softw. 157 (2019) 110389.
[26] Kamonphop Srisopha, Barry Boehm, Pooyan Behnamghader, Do consumers talk about the software in my product? an exploratory study of iot products on amazon, CLEI Electron. J. 22 (04) (2019).
[27] Luis Oliveira, Val Mitchell, Andrew May, Smart home technology?comparing householder expectations at the point of installation with experiences 1 year later, Pers. Ubiquitous Comput. 24 (5) (2020) 613–626.
[28] Serena Zheng, Noah Apthorpe, Marshini Chetty, Nick Feamster, User perceptions of smart home IoT privacy, Proc. ACM on Human–Comput. Interact. 2 (CSCW) (2018) 1–20.
[29] Forrest Shull, Janice Singer, Dag I.K. Sjøberg, Guide To Advanced Empirical Software Engineering, Springer, 2007.
[30] Claes Wohlin, Per Runeson, Martin Höst, Magnus C Ohlsson, Björn Regnell, Anders Wesslén, Experimentation in Software Engineering, Springer Science & Business Media, 2012.
[31] StatCounter, Mobile Operating System Market Share Worldwide - 2022, 2022, https://gs.statcounter.com/os-market-share/mobile/worldwide.
[32] Shuyo Nakatani, Language Detection Library for Java, 2010, https://github.com/shuyo/language-detection.
[33] Bo Pang, Lillian Lee, et al., Opinion mining and sentiment analysis, Found. Trends® in Inf. Retrieval 2 (1–2) (2008) 1–135.
[34] Clayton Hutto, Eric Gilbert, Vader: A parsimonious rule-based model for sentiment analysis of social media text, in: Proceedings of the International AAAI Conference on Web and Social Media, vol. 8, 2014, pp. 216–225.
[35] Mike Thelwall, The heart and soul of the web? Sentiment strength detection in the social web with SentiStrength, in: Cyberemotions, vol. 11, Springer, 2017, pp. 9–134.
[36] Christopher D. Manning, Mihai Surdeanu, John Bauer, Jenny Rose Finkel, Steven Bethard, David McClosky, The Stanford CoreNLP natural language processing toolkit, in: Proceedings of 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations, 2014, pp. 55–60.
[37] Filipe N. Ribeiro, Matheus Araújo, Pollyanna Gonçalves, Marcos André Gonçalves, Fabrício Benevenuto, Sentibench-a benchmark comparison of state-of-the-practice sentiment analysis methods, EPJ Data Sci. 5 (1) (2016) 1–29.
[38] Jacek Dąbrowski, Emmanuel Letier, Anna Perini, Angelo Susi, Analysing app reviews for software engineering: A systematic literature review, Empirical Softw. Engg. 27 (2) (2022) 63, http://dx.doi.org/10.1007/s10664-021-10065-7.
[39] Ning Chen, Jialiu Lin, Steven C.H. Hoi, Xiaokui Xiao, Boshen Zhang, AR-miner: Mining informative reviews for developers from mobile app marketplace, in: Proceedings of the 36th International Conference on Software Engineering (ICSE 2014), 2014, pp. 767–778, http://dx.doi.org/10.1145/2568225.2568263.
[40] Stuart Mcilroy, Nasir Ali, Ahmed E. Hassan, Fresh apps: An empirical study of frequently-updated mobile apps in the google play store, Empirical Softw. Engg. 21 (3) (2016) 1346–1370, http://dx.doi.org/10.1007/s10664-015-9388-2.
[41] Henry B. Mann, Donald R. Whitney, On a test of whether one of two random variables is stochastically larger than the other, Ann. Math. Stat. (1947) 50–60.
[42] Klaus Krippendorff, Content Analysis: An Introduction To Its Methodology, Sage publications, 2018.
[43] Klaus Krippendorff, Reliability in content analysis: Some common misconceptions and recommendations, Hum. Commun. Res. 30 (3) (2004) 411–433.
[44] Johnny Saldaña, The Coding Manual for Qualitative Researchers, sage, 2021.
[45] Nan Hu, Jie Zhang, Paul A. Pavlou, Overcoming the J-shaped distribution of product reviews, Commun. ACM 52 (10) (2009) 144–147.
[46] Lili Wei, Yepang Liu, Shing-Chi Cheung, Taming android fragmentation: Characterizing and detecting compatibility issues for android apps, in: Proceedings of the 31st IEEE/ACM International Conference on Automated Software Engineering, 2016, pp. 226–237.
[47] Colin M. Gray, Yubo Kou, Bryan Battles, Joseph Hoggatt, Austin L. Toombs, The dark (patterns) side of UX design, in: Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems, 2018, pp. 1–14.
[48] Arvind Narayanan, Arunesh Mathur, Marshini Chetty, Mihir Kshirsagar, Dark patterns: Past, present, and future: The evolution of tricky user interfaces, Queue 18 (2) (2020) 67–92.
[49] Cherie Lacey, Catherine Caudwell, Cuteness as a dark pattern in home robots, in: 2019 14th ACM/IEEE International Conference on Human-Robot Interaction, HRI, IEEE, 2019, pp. 374–381.
[50] Preksha Nema, Pauline Anthonysamy, Nina Taft, Sai Teja Peddinti, Analyzing user perspectives on mobile app privacy at scale, in: 2022 IEEE/ACM 44rd International Conference on Software Engineering, ICSE, IEEE, 2022.
[51] Duc Cuong Nguyen, Erik Derr, Michael Backes, Sven Bugiel, Short text, large effect: Measuring the impact of user reviews on android app security & privacy, in: 2019 IEEE Symposium on Security and Privacy, SP, IEEE, 2019, pp. 555–569.
[52] Gian Luca Scoccia, Stefano Ruberto, Ivano Malavolta, Marco Autili, Paola Inverardi, An investigation into Android run-time permissions from the end users' perspective, in: Proceedings of the 5th International Conference on Mobile Software Engineering and Systems, 2018, pp. 45–55.
[53] David L. Parnas, A John Van Schouwen, Shu Po Kwan, Evaluation of safety-critical software, Commun. ACM 33 (6) (1990) 636–648.
[54] João Pedro Dias, Flávio Couto, Ana C.R. Paiva, Hugo Sereno Ferreira, A brief overview of existing tools for testing the internet-of-things, in: 2018 IEEE International Conference on Software Testing, Verification and Validation Workshops, ICSTW, IEEE, 2018, pp. 104–109.
[55] Erik Jan Marinissen, Yervant Zorian, Mario Konijnenburg, Chih-Tsun Huang, Ping-Hsuan Hsieh, Peter Cockburn, Jeroen Delvaux, Vladimir Rožić, Bohan Yang, Dave Singelée, et al., IoT: Source of test challenges, in: 2016 21th IEEE European Test Symposium, ETS, IEEE, 2016, pp. 1–10.
[56] Bestoun S. Ahmed, Miroslav Bures, Karel Frajtak, Tomas Cerny, Aspects of quality in Internet of Things (IoT) solutions: A systematic mapping study, IEEE Access 7 (2019) 13758–13780.
[57] Abbas Ahmad, Fabrice Bouquet, Elizabeta Fourneret, Franck Le Gall, Bruno Legeard, Model-based testing as a service for iot platforms, in: Leveraging Applications of Formal Methods, Verification and Validation: Discussion, Dissemination, Applications: 7th International Symposium, ISoLA 2016, Imperial, Corfu, Greece, October (2016) 10-14, Proceedings, Part II 7, Springer, 2016, pp. 727–742.
[58] Philipp Rosenkranz, Matthias Wählisch, Emmanuel Baccelli, Ludwig Ortmann, A distributed test system architecture for open-source IoT software, in: Proceedings of the 2015 Workshop on IoT Challenges in Mobile and Industrial Systems, 2015, pp. 43–48.

[59] Harald Foidl, Michael Felderer, Data science challenges to improve quality assurance of Internet of Things applications, in: Leveraging Applications of Formal Methods, Verification and Validation: Discussion, Dissemination, Applications: 7th International Symposium, ISoLA 2016, Imperial, Corfu, Greece, October (2016) 10-14, Proceedings, Part II 7, Springer, 2016, pp. 707–726.
[60] Tao Wang, Kangkang Zhang, Wei Chen, Wensheng Dou, Jiaxin Zhu, Jun Wei, Tao Huang, Understanding device integration bugs in smart home system, in: Proceedings of the 31st ACM SIGSOFT International Symposium on Software Testing and Analysis, 2022, pp. 429–441.
[61] Konstantinos Arakadakis, Pavlos Charalampidis, Antonis Makrogiannakis, Alexandros Fragkiadakis, Firmware over-the-air programming techniques for IoT networks-a survey, ACM Comput. Surv. 54 (9) (2021) 1–36.
[62] Dayi Lin, Cor-Paul Bezemer, Ahmed E. Hassan, Studying the urgent updates of popular games on the steam platform, Empir. Softw. Eng. 22 (2017) 2095–2126.
[63] Filipe R. Cogo, Gustavo A. Oliva, Cor-Paul Bezemer, Ahmed E. Hassan, An empirical study of same-day releases of popular packages in the npm ecosystem, Empir. Softw. Eng. 26 (5) (2021) 89.
[64] Saad El Jaouhari, Eric Bouvet, Secure firmware over-the-air updates for IoT: Survey, challenges, and discussions, Internet of Things 18 (2022) 100508.
[65] Bixin Li, Xiaobing Sun, Hareton Leung, Sai Zhang, A survey of code-based change impact analysis techniques, Softw. Test. Verif. Reliab. 23 (8) (2013) 613–646.
[66] Francisco Javier Acosta Padilla, Emmanuel Baccelli, Thomas Eichinger, Kaspar Schleiser, The future of IoT software must be updated, in: IAB Workshop on Internet of Things Software Update (IoTSU), 2016.
[67] Maleknaz Nayebi, Bram Adams, Guenther Ruhe, Release practices for mobile apps–what do users and developers think? in: 2016 Ieee 23rd International Conference on Software Analysis, Evolution, and Reengineering (Saner), vol. 1, IEEE, 2016, pp. 552–562.
[68] Chola Chhetri, Vivian Genaro Motti, Eliciting privacy concerns for smart home devices from a user centered perspective, in: International Conference on Information, Springer, 2019, pp. 91–101.
[69] Pardis Emami-Naeini, Yuvraj Agarwal, Lorrie Faith Cranor, Hanan Hibshi, Ask the experts: What should be on an IoT privacy and security label? in: 2020 IEEE Symposium on Security and Privacy (SP), IEEE, 2020, pp. 447–464.
[70] Pardis Emami-Naeini, Janarth Dheenadhayalan, Yuvraj Agarwal, Lorrie Faith Cranor, An informative security and privacy nutrition label for Internet of Things devices, IEEE Secur. Privacy 20 (2) (2021) 31–39.
[71] Kim J. Kaaz, Alex Hoffer, Mahsa Saeidi, Anita Sarma, Rakesh B. Bobba, Understanding user perceptions of privacy, and configuration challenges in home automation, in: 2017 IEEE Symposium on Visual Languages and Human-Centric Computing (VL/HCC), IEEE, 2017, pp. 297–301.
[72] Paolo Calciati, Konstantin Kuznetsov, Alessandra Gorla, Andreas Zeller, Automatically granted permissions in android apps: An empirical study on their prevalence and on the potential threats for privacy, in: Proceedings of the 17th International Conference on Mining Software Repositories, 2020, pp. 114–124.
[73] Gary Liu, Nathan Malkin, Effects of privacy permissions on user choices in voice assistant app stores, Proc. Privacy Enhancing Technol. 4 (2022) 421–439.
[74] Nathan Malkin, David Wagner, Serge Egelman, Runtime permissions for privacy in proactive intelligent assistants, in: Eighteenth Symposium on Usable Privacy and Security (SOUPS 2022), 2022, pp. 633–651.
[75] Chola Chhetri, Vivian Genaro Motti, User-centric privacy controls for smart homes, Proc. ACM on Human-Comput. Interact. 6 (CSCW2) (2022) 1–36.
[76] Maximilian Maier, Rikard Harr, Dark design patterns: An end-user perspective, Hum. Technol. 16 (2) (2020) 170.
[77] Linda Di Geronimo, Larissa Braz, Enrico Fregnan, Fabio Palomba, Alberto Bacchelli, UI dark patterns and where to find them: a study on mobile applications and user perception, in: Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems, 2020, pp. 1–14.
[78] Colin M. Gray, Jingle Chen, Shruthi Sai Chivukula, Liyang Qu, End user accounts of dark patterns as felt manipulation, Proc. ACM on Human-Comput. Interact. 5 (CSCW2) (2021) 1–25.
[79] Kerstin Bongard-Blanchy, Arianna Rossi, Salvador Rivas, Sophie Doublet, Vincent Koenig, Gabriele Lenzini, I am definitely manipulated, even when I am aware of it. It's ridiculous!-dark patterns from the end-user perspective, in: Designing Interactive Systems Conference 2021, 2021, pp. 763–776.
[80] Thomas D. Cook, Donald Thomas Campbell, Arles Day, Quasi-Experimentation: Design & Analysis Issues for Field Settings, vol. 351, Houghton Mifflin Boston, 1979.
[81] W. Maalej, H. Nabil, Bug report, feature request, or simply praise? On automatically classifying app reviews, in: 2015 IEEE 23rd International Requirements Engineering Conference, RE, 2015, pp. 116–125, http://dx.doi.org/10.1109/RE.2015.7320414.
[82] Andrea Di Sorbo, Sebastiano Panichella, Carol V. Alexandru, Junji Shimagaki, Corrado A. Visaggio, Gerardo Canfora, Harald C. Gall, What would users change in my app? Summarizing app reviews for recommending software changes, in: Proceedings of the 2016 24th ACM SIGSOFT International Symposium on Foundations of Software Engineering (FSE 2016), 2016b, pp. 499–510, http://dx.doi.org/10.1145/2950290.2950299.
[83] Jianmao Xiao, Shizhan Chen, Qiang He, Hongyue Wu, Zhiyong Feng, Xiao Xue, Detecting user significant intention via sentiment-preference correlation analysis for continuous app improvement, in: Eleanna Kafeza, Boualem Benatallah, Fabio Martinelli, Hakim Hacid, Athman Bouguettaya, Hamid Motahari (Eds.), Service-Oriented Computing, 2020, pp. 386–400.
[84] Bilge Celik, Joaquin Vanschoren, Adaptation strategies for automated machine learning on evolving data, IEEE Trans. Pattern Anal. Mach. Intell. 43 (9) (2021) 3067–3078, http://dx.doi.org/10.1109/TPAMI.2021.3062900.