Interdisciplinary Journal of Information, Knowledge, and Management

*An Official Publication of the Informing Science Institute InformingScience.org*

*IJIKM.org*

**Volume 20, 2025**

# FROM CONVENTIONAL METHODS TO LARGE LANGUAGE MODELS: A SYSTEMATIC REVIEW OF TECHNIQUES IN MOBILE APP REVIEW ANALYSIS

| | | |
|---|---|---|
| Nimasha Arambepola* | Software Engineering Teaching Unit, Faculty of Science, University of Kelaniya, Sri Lanka | nimasha@kln.ac.lk |
| Lankeshwara Munasinghe | School of Computing, Engineering, and Technology, Robert Gordon University, Aberdeen, Scotland, United Kingdom | l.munasinghe@rgu.ac.uk |
| Waruni Wimalasena | Software Engineering Teaching Unit, Faculty of Science, University of Kelaniya, Sri Lanka | rawlw241@kln.ac.lk |

* Corresponding author

## ABSTRACT

| | |
|---|---|
| Aim/Purpose | This paper focuses on app review analysis techniques, driven by the rapid advancement of the mobile app market and NLP techniques in optimizing mobile app user experiences. |
| Background | Owing to technological advancements, app review analysis has rapidly evolved. This study examines both conventional and emerging techniques, including current advancements such as large language models (LLMs) in app review analysis. It provides an overview of the various methods used across different categories of app review analysis, comparing effective strategies for identifying user concerns and enhancing app functionality. |
| Methodology | A systematic review was utilized based on two major standard guidelines, PRISMA and Kitchenham's guidelines, for the period of 2014 to 2024. After defining the review protocol, papers were identified through keyword-based searches on six major online databases: Scopus, Web of Science, IEEE Xplore, ACM Digital Library, Science Direct, and Springer. Following screening and excluding papers based on defined quality criteria, 53 papers were considered for this study. The use of |

Accepting Editor Ahmad Samed Al-Adwan | Received: November 20, 2024 | Revised: March 5, March 27, April 14, 2025 | Accepted: April 15, 2025.
Cite as: Arambepola, N., Munasinghe, L., & Wimalasena, W. (2025). From conventional methods to large language models: A systematic review of techniques in mobile app review analysis. *Interdisciplinary Journal of Information, Knowledge, and Management, 20,* Article 16. https://doi.org/10.28945/5491

PRISMA ensures a transparent and reproducible review process, while Kitchenham's guidelines provide a structured and rigorous approach for evaluating and synthesizing the literature.

Contribution — This review study aims to evaluate the current state of knowledge on app review analysis techniques to improve mobile app user experiences. This study categorized the existing state-of-the-art papers into eight different categories, such as sentiment analysis, review classification, summarization, and prioritization, and examined challenges related to app review analysis. Furthermore, the study emphasizes the potential of LLMs for optimizing and automating app review analysis and provides future directions to address gaps in user-centric app development.

Findings — Among the eight main categories defined in app review analysis, sentiment analysis is the most prevalent, followed by review classification and information extraction. Most studies use a combination of these categories to achieve a comprehensive goal. Prioritization techniques such as risk matrices, thumbs-up count-based approach, and anomaly detection are widely used to identify emerging issues. Extracting meaningful information and evaluating the proposed approach are the most common challenges identified. Novel LLMs, like Chat-GPT, significantly enhance review analysis by automating the process, improving feature extraction, and enabling context-aware review classification.

Recommendations for Practitioners — The combination of conventional approaches and novel LLM-based methods can enhance both the efficiency and accuracy in identifying and addressing critical issues raised through mobile app user reviews. It effectively prioritizes user concerns by leveraging the strengths of both traditional preprocessing techniques and advanced LLMs.

Recommendations for Researchers — Researchers are encouraged to explore the integration of emerging technologies like LLMs to enhance the of app review analysis, particularly in feature-specific sentiment analysis.

Impact on Society — The results of this study contribute to enhancing the mobile app user experience through effective app review analysis, which improves user satisfaction and supports user-centered app development. This ultimately leads to a better mobile app ecosystem, benefiting both users and developers.

Future Research — In the future, this research can be extended in multiple directions. Researchers can address the existing research gaps that LLMs have yet to address, particularly in prioritizing user concerns. Additionally, there is potential for further research on tool implementations focusing on identifying persistent issues through time series analysis by considering the app version and date of the app reviews. Moreover, there is a need to develop comprehensive frameworks that are more generalizable across different apps and categories, with a focus on identifying user concerns related to specific features.

Keywords — app user reviews, text analysis, LLM, user experience, systematic review

## INTRODUCTION

Owing to the widespread adoption of smartphones and the increasing dependence on mobile applications, the mobile app market has seen significant expansion throughout the past decade. Especially during and after the COVID-19 pandemic, people established and maintained their daily activities via

mobile applications (Chemnad et al., 2022). For example, several new mobile apps have been introduced to the app market, mainly in app categories such as education, social media, lifestyle, shopping, and entertainment. This expansion motivates continuous and rapid app enhancement to retain and attract users. In this competitive app market, app user reviews are a crucial feedback mechanism that provides direct insights into users' experiences, preferences, expectations, and difficulties while using the app.

App review analysis involves extracting meaningful information from user feedback to identify common issues, desired features, and overall user satisfaction. This process is essential for prioritizing user requirements to make decisions for the next release plan to optimize the user experience (UX). App review analysis is carried out in different categories/types, including sentiment analysis, review classification, and review prioritization (Sultana & Sarker, 2018; Villarroel et al., 2016). However, manual analysis is impractical for millions of reviews. Therefore, it is necessary to employ techniques such as machine learning (ML) and natural language processing (NLP).

Despite ML and NLP, large language models (LLMs) have emerged as powerful tools in recent years, and their popularity has grown exponentially. The ability of LLMs for sentiment analysis has been emerging as a promising area across various industries, including e-commerce, social media monitoring, healthcare, and finance, to gain insights about customer opinions, brand perception, market trends, and public sentiment, enabling data-driven decision-making and enhanced customer experiences (Upadhye, 2024). For example, ChatGPT, a state-of-the-art LLM, has shown its potential to conduct advanced aspect-based analysis of hotel reviews to understand the areas lacking customer satisfaction (Jeong & Lee, 2024). With their content extraction and prompt generation capabilities, LLMs have also shown their capabilities in app testing by helping detect unusual input detection (Z. Liu et al., 2024). In addition to these capabilities, LLMs have proven instrumental in analyzing macro and micro-level customer behavior trends over time to measure the impact of negative reviews on consumer behavior (Z. Wang et al., 2024). The growth of LLMs' capabilities also expands into academic research areas, especially in the systematic literature review (SLR) process, which starts by automating the screening process of SLR (Dennstädt et al., 2024). Then, it expanded into automating the initial search, screening, summarization, and analysis phases in the manual SLR (Sami et al., 2024). These approaches prove the capability of LLMs to evolve the manual analyzing processes into a much more efficient and effective automated process.

The use of LLMs has also brought about notable advancements in the efficiency of app review analysis. With the popularity of LLMs such as Chat-GPT, Gemini, and LLaMA, the capability of automating review analysis, information extraction, and response generation has been explored in a new aspect of this area. These models are designed to capture subtle linguistic nuances, enabling them to provide more accurate and comprehensive analyses (Roumeliotis et al., 2024). It has been widely spread across various aspects, including improving the app review process and e-commerce platforms (Azov et al., 2024) and hotel services (Jeong & Lee, 2024) by analyzing customer comments, which are much more relatable to app reviews. LLMs' improvements offer enhanced capabilities for review classification, sentiment analysis, and feature extraction (Assi et al., 2024; Zhang et al., 2024). They can operate in complex scenarios, such as extracting user sentiments and app features with varied levels of contextual understanding (Roumeliotis et al., 2024; Shah et al., 2024). For instance, ChatGPT and GPT-4 models have effectively analyzed feedback to identify user preferences and complaints (Roumeliotis et al., 2024; Shah et al., 2024). Furthermore, approaches such as LLM-Cure utilize LLMs in multiple phases, like categorizing feedback, identifying underperforming features, and suggesting improvements by referencing highly rated features from competitors (Assi et al., 2024). This proves that LLMs demonstrate the versatile applications of these models in app review analysis, allowing stakeholders to pinpoint issues, prioritize user concerns, and align development strategies to optimize user experience effectively (Assi et al., 2024; Zhao et al., 2024).

Despite their benefits, LLMs still have many limitations. Challenges related to data privacy, contextual understanding, and handling ambiguous feedback remain areas of ongoing research (Zhao et al.,

2024). Additionally, while LLMs perform well in conventional tasks such as sentiment classification and fundamental text analysis, they may struggle with more complex tasks that require deep contextual or structured sentiment information (Morbidoni, 2023; Zhang et al., 2024). For instance, complex tasks like quadruple extraction, which require specialized prompt templates and methods to leverage few-shot examples, test the capability boundaries of LLMs (de Lima et al., 2023b). Comparative evaluations against state-of-the-art models on public datasets have shown both the potential and limitations of LLMs in handling such complex extractions (Xu et al., 2023). Future advancements in LLM training, domain-specific fine-tuning (Roumeliotis et al., 2024), and combining LLMs with retrieval-augmented generation (Azov et al., 2024) are expected to address these challenges. Such improvements will enhance the ability of app developers and researchers to leverage LLMs for continuous, scalable, and accurate app review analysis and feature enhancement. Continued research refining LLMs' contextual comprehension and domain adaptability will further improve their application in app review analysis, ultimately focusing on more user-centered, innovative, and competitive mobile app experiences. Despite the advancement, some methods, such as review labelling and topic evaluation, still require manual effort. Even though generative artificial intelligence (AI) is capable of automating text labelling, it still relies on manually labelled datasets to improve precision and relevance (Chou & Cho, 2023; Kozlowski et al., 2024). Therefore, it remains a challenge to conduct a fully automatic review analysis.

Review analysis is a well-established yet continuously evolving research field. Researchers in this field utilize different and combined techniques to develop frameworks that enhance accuracy and efficiency in review analysis. Consequently, despite the advancements in user review analysis, several challenges still exist. Therefore, it is beneficial to identify different challenges and limitations with different techniques and identify methods to overcome them for future research studies. Thus, this systematic review provides a comprehensive overview of state-of-the-art techniques in app review analysis and challenges for app review analysis in software engineering by examining previous research studies published between 2014 and 2024 across six major academic databases. While previous review studies have addressed app review analysis (Dąbrowski et al., 2022; Genc-Nayebi & Abran, 2017), this study identifies explicitly gaps in prioritization methods for user concerns, examines challenges in app review analysis, and explores potential strategies to overcome them.

There are several notable gaps in the existing systematic reviews on app review analysis. A significant gap is the lack of focus on prioritization methods for user concerns. While many reviews emphasize sentiment analysis and classification (Dąbrowski et al., 2022; Genc-Nayebi & Abran, 2017), they neglect how to align user feedback with development priorities. This limits developers' guidance on decision-making. Moreover, there is an insufficient exploration of emerging techniques, such as LLMs. Additionally, despite the growing need for responsive and agile development, cross-domain applications and real-time analysis are often neglected. This study seeks to address these gaps by reviewing app review prioritization techniques, LLM integration, and cross-domain application review analysis. Therefore, the objectives of this systematic review are to categorize types of app review analysis, identify various techniques, including emerging LLM-based methods, examine prioritization approaches, and analyze the challenges and solutions in app review analysis.

The remainder of this paper is organized as follows. The next section discusses the review process and methodology used for the analysis in this study. Then, the results of the findings are presented and discussed. The paper concludes with a discussion of future research directions.

## RELATED WORKS

Over the past decade, user reviews have become more complex due to the increasing diversity of app users and their evolving requirements. Mobile app review analysis aims to enhance UX by understanding user feedback. However, due to many lengthy, non-informative, erroneous user reviews, it is

difficult for users and developers to read every review to understand user concerns related to a particular app. Even though an app rating system is available to express the overall user opinion, disparities exist between user ratings and review comments. Consequently, a sentiment rating approach has been proposed to provide summarised feedback, providing users with a clearer understanding of the application beyond the star rating (Rodrigues et al., 2017; Yu et al., 2017).

## *Challenges in App Review Analysis*

There are common challenges for app review analysis due to inconsistencies in app user reviews. Among them, many existing studies highlight the need for effective preprocessing techniques due to the difficulties associated with extracting meaningful information from app reviews. Early works in this field typically focused on basic text preprocessing techniques. For instance, tokenization, stop word removal, stemming, and lemmatization are the most common initial preprocessing steps in text analysis (Genc-Nayebi & Abran, 2017). These methods laid the foundation for app review analysis, but they were limited in their ability to handle the complexities of user feedback. For example, tokenization and stemming frequently ignore the context in which words are used, potentially resulting in the loss of important meaning. Specifically, this affects the reviews that contain domain-specific terms (Gao et al., 2022). Additionally, non-informative reviews can be categorized as meaningful and meaningless reviews. While meaningless, non-informative reviews are not helpful, meaningful non-informative reviews can be useful for initial filtering and primary analyses. For example, a review such as "Really nice app" is helpful for sentiment analysis to determine a positive or negative opinion, but it lacks details in identifying specific aspects of the app that are appreciated or need improvement. Thus, this is non-informative for extracting meaningful insights. Filtering and extracting only meaningful sentences by setting a threshold for review length can overcome the issue of having a sheer volume of non-informative reviews in the review analysis. Furthermore, custom stop word removal has been widely used, as certain words are meaningless for identifying prominent topics or themes from app user reviews (Arambepola et al., 2024; Gao et al., 2022). Despite this, many studies still overlook the importance of handling mixed or neutral sentiment reviews, which often lack clear polarity but may contain significant information about app features that need improvement.

## *Applications of App Review Analysis*

App review analyses have been conducted for various purposes, with some studies explicitly focusing on particular app categories, such as health and fitness (Ahn & Park, 2023; Haggag et al., 2022). The findings from these analyses are integral at various stages of the software development life cycle, from requirement gathering to app maintenance (Al-Subaihin et al., 2021; Dąbrowski et al., 2022, 2023). Consequently, researchers have conducted app review analysis to identify the supporting software engineering activities and investigate user reviews related to specific aspects of apps. For example, usability and user experience identification through app reviews is widely adopted (Lim et al., 2021; W. Nakamura et al., 2022), particularly emphasizing user interface improvements (Q. Chen et al., 2021). Moreover, apps that satisfy users in some countries may not meet users' expectations in other countries due to economic disparities and different user expectations (Srisopha et al., 2020). Thus, country-specific feature requests are essential for customizing mobile apps based on the preferences of user demographics.

Furthermore, app review analysis is crucial for market research for app development, as it allows comparing competitive mobile apps in app stores (Li et al., 2017). In there, feature-oriented sentiment analysis is vital for understanding which features contribute positively and negatively to user experience (Luiz et al., 2018). This informative classification also supports decision-making for upcoming app release updates and patches. Different tools and frameworks have been proposed to assist in analyzing app reviews. For instance, AR-Miner is a tool that is used to gather feedback from users (N. Chen et al., 2014), SUR miner permits sentiment analysis together with topic modelling (Di Sorbo et al., 2016), while MApp-IDEA is an involved review analytics and data visualization platform

(Gao et al., 2018). These advancements show significant progress in app review analysis. However, handling many heterogeneous reviews characterized by context-richness still needs to be solved.

## *State-of-the-Art Techniques in App Review Analysis*

Recent studies underscore the significance of large language models (LLMs) in user review analysis across various domains. Automated feature-level sentiment analysis in app reviews has proven essential, with state-of-the-art LLMs, such as GPT-4, surpassing rule-based methods in extracting feature-specific sentiments and generating valuable insights from minimal labeled data (Shah et al., 2024). Similarly, LLMs play a crucial role in e-commerce by enhancing customer sentiment analysis through comparisons of fine-tuned and pre-trained models, contributing to a deeper understanding of user satisfaction and improving customer experience(Roumeliotis et al., 2024). These findings affirm the potential of LLMs in automated review analysis, extending their utility beyond conventional sentiment classification.

Recent research has focused on LLMs for more advanced and automated app review analyses. The emergence of LLM-based tools, such as LLM-Cure, has improved more targeted and effective feature enhancement suggestions by identifying underperforming app features and comparing them with highly rated features from competing applications (Assi et al., 2024). This tool emphasizes the potential of LLMs in augmenting app review analysis, automating feature extraction, and generating informed recommendations for product enhancement. Another significant contribution is the application of LLMs for structured user review analysis, which employs retrieval-augmented generation (RAG) for more accurate data processing (Azov et al., 2024). New systems like SCRABLE have been designed to self-optimize their prompts and assess response quality using an LLM-based judging mechanism. The efficacy of such systems is evidenced by their ability to produce high-quality responses that exceed baseline results by substantial margins. This reinforces the importance of combining advanced LLM capabilities with adaptive and context-aware mechanisms to enhance the accuracy and efficiency of app review analysis (Azov et al., 2024). Moreover, integrating LLMs into the app review analysis process supports a more holistic understanding of user feedback. Unlike traditional methods that rely on primary sentiment classification, LLMs facilitate aspect-based sentiment analysis. They can process complex quadruple extractions using specialized prompt templates for aspect, category, opinion, and sentiment extraction to identify opinions related to specific app features (Xu et al., 2023). Recent research has also introduced a dynamic prompt generation technique to extract specific application characteristics from user reviews and classify risks by severity, from negligible to critical. This approach enables the automatic construction of a standardized risk matrix, with evidence showing that the Open Pre-trained Transformers (OPT) model competes well with proprietary models like GPT-3.5 (de Lima et al., 2023b).

Adopting LLMs has transformed app review analysis, shifting it from conventional sentiment classification to more sophisticated, context-rich evaluations that can inform actionable strategies. The future of this field will likely see further integration of LLMs with user-contributed documents and real-time data processing to create dynamic, continuously learning systems that better cater to user and developer needs. This evolution aligns with the broader trend of AI-driven decision-making in software development. This has the potential to position LLMs as crucial tools in extracting, interpreting, and acting on user feedback to drive competitive advantage and user satisfaction.

## *Ethical Considerations and Limitations*

While LLMs offer significant potential for app review analysis, their deployment raises several ethical and practical concerns. A major limitation is the representativeness of the datasets used to train these models, which may not fully capture the diversity of user experiences, thereby affecting their generalizability across different application domains (Roumeliotis et al., 2024). Additionally, privacy and data security concerns remain paramount, as the automated processing of user reviews involves handling sensitive information. Addressing these challenges necessitates the implementation of robust ethical

frameworks and regulatory guidelines to ensure transparency and accountability in LLM-based analysis (Zhao et al., 2024).

Moreover, the responsible use of sentiment analysis systems requires careful consideration of potential biases, societal impacts, and unintended consequences. Ethical guidelines and stakeholder engagement mechanisms are essential to mitigate risks associated with automated decision-making and ensure the fair and responsible deployment of AI-driven review analysis tools (Upadhye, 2024). Furthermore, the evolving nature of LLM ecosystems calls for continuous dialogue between developers, policymakers, and researchers to establish best practices for responsible AI development, particularly in relation to security, privacy, and user trust (Zhao et al., 2024). By incorporating these ethical considerations, future research can contribute to developing more transparent, accountable, and socially responsible LLM-based review analysis systems.

### *Recent Research Trends and Gaps*

Despite advancements in app review analysis, several critical gaps remain unaddressed. One significant limitation is the lack of emphasis on prioritization strategies for user concerns. Existing studies primarily focus on sentiment analysis and classification (Dąbrowski et al., 2022; Genc-Nayebi & Abran, 2017) yet fail to establish methodologies for aligning user feedback with development priorities. This shortcoming restricts developers' ability to make informed decisions based on user needs. Additionally, there is limited exploration of emerging approaches, particularly the integration of LLMs, in refining app review analysis (Noei & Lyons, 2019). Furthermore, despite the growing demand for agile and responsive development practices, cross-domain applications and real-time analytics remain underexplored. Addressing these gaps is essential for enhancing the effectiveness of app review analysis. This study seeks to bridge these deficiencies by investigating prioritization frameworks, LLM-driven review synthesis, and methodologies for analyzing cross-domain app reviews in real time.

## METHODOLOGY

A systematic review collects and synthesizes findings to address specific questions within a given field or subject. Several standard guidelines for performing systematic literature reviews vary depending on the area or domains involved. For instance, commonly used guidelines include Kitchenham's guidelines (Kitchenham & Charters, 2007) and the Preferred Reporting Items for Systematic Reviews and Meta-Analyses (PRISMA) (Page et al., 2021). Although PRISMA primarily applies to healthcare, social science, and educational research, it is also popular in computer science research (Boaye Belle & Zhao, 2022). In contrast, Kitchenham's guidelines provide comprehensive instructions for systematic literature reviews in Software Engineering. Therefore, this study adopts a combined approach, utilizing both guidelines for this systematic literature review.

The study was conducted in three stages: planning, conducting, and reporting the review. The most crucial stage, planning, involved formulating the research questions and developing and evaluating the review protocol. The review protocol consists of the following components: search strategy (keywords and databases), study inclusion and exclusion criteria, and data extraction approach (summarizing and analyzing data). The review protocol was assessed using a set of pre-selected studies pertinent to the research area. Subsequently, the review was conducted according to the finalized review protocol. The main stages are discussed in the following sections.

### *Formulating Research Questions*

Formulating the research question is a critical stage in a systematic review as it sets the direction for the study. Therefore, it is essential to establish clear and concise research questions that align with the overall research objective. Thus, the formulated research questions are as follows:

**RQ1:** What are the main categories of app review analysis in Software Engineering?

**RQ2:** Which techniques are employed to perform app review analyses?

**RQ3:** What are the techniques used for prioritization of user concerns?

**RQ4:** What challenges are encountered in app review analysis?

**RQ5:** How have LLMs evolved for the app review analysis process?

A systematic literature review followed the developed review protocol to address these research questions.

## *Literature Search and Selection*

The PRISMA method was followed to select literature for the review, with publications from 2014 to 2024 being considered. Initially, papers were identified through keyword-based searches on six major online databases: Scopus, Web of Science, IEEE Xplore, ACM Digital Library, Science Direct, and Springer. The key terms used for the search were 'app review analysis,' 'app review mining,' 'analyzing app reviews,' and 'mining app reviews.' As shown in Figure 1, a total of 221 research papers were identified from the initial search. According to the diagram, after removing 56 duplicate research articles, 165 articles were screened according to the inclusion and exclusion criteria stated in Table 1. As the research scope was narrowed, most of the identified research articles were excluded due to their lack of relevance to the study. For instance, 116 publications were excluded based on the inclusion and exclusion criteria.

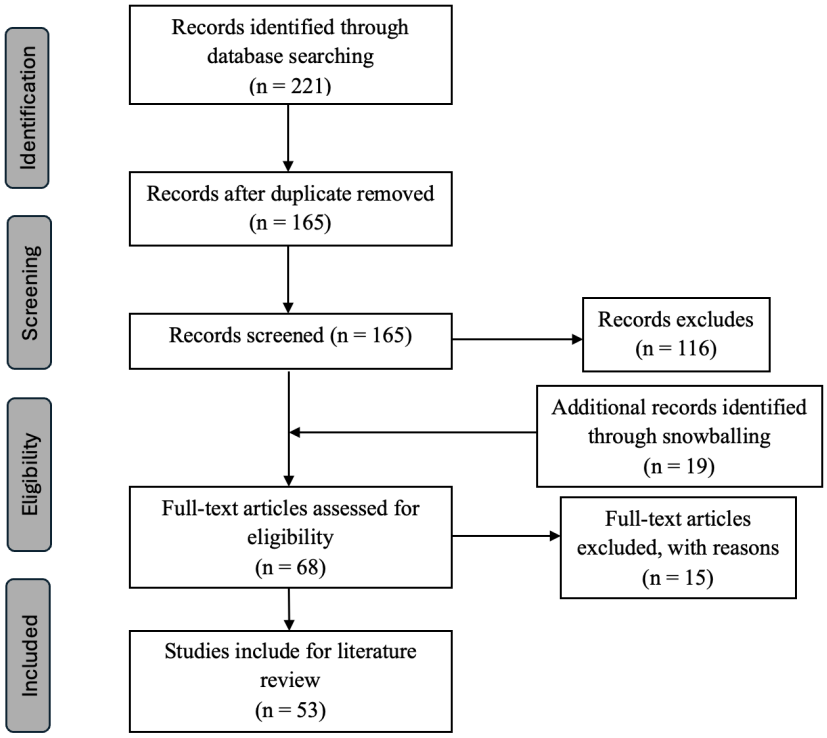


**Figure 1. PRISMA diagram of study selection**

Consequently, backward and forward snowballing was performed, starting with the 12 most relevant research papers, which led to the identification of 19 further relevant studies for the literature review. The articles for the literature review were then finalized, with additional exclusions based on quality assessment criteria. The quality of the primary studies was evaluated using the checklist shown in Table 2. Among the questions in the quality checklist, only papers that adhered to at least four questions were considered for the review. Ultimately, a total of 53 papers were considered for the review.

**Table 1. Study selection criteria (inclusion and exclusion criteria)**

| Number | Inclusion criteria |
|---|---|
| **1** | Studies published as conference papers, journal articles, or book chapters as full papers. |
| **2** | Studies related to app review analysis that support software engineering activities, either directly or indirectly. |
| **Number** | **Exclusion criteria** |
| **1** | Papers not written in English. |
| **2** | Studies not published as full papers. |
| **3** | Studies not related to the specified research questions. |

**Table 2. Quality checklist (Kitchenham & Charters, 2007)**

| Number | Question |
|---|---|
| **1** | Are the objectives clearly defined? |
| **2** | How clear and transparent is the method of evaluative assessment? |
| **Number** | **Exclusion criteria** |
| **3** | How effectively was data gathering conducted? |
| **4** | How clearly has the approach to and description of analysis been presented? |
| **5** | How clear are the links between data, interpretation, and conclusions? |
| **6** | How adequately has the research process been documented? |

## *Data Extraction and Synthesis*

The data extraction approach was established during the review protocol development stage. The majority of the selected research papers have used a mix of qualitative and quantitative methods. Data from the selected papers was extracted using an electronic spreadsheet across ten specified fields, focusing on addressing the formulated research questions. The extracted data was organized in the spreadsheet for easy identification, aggregation, and comparison (Dąbrowski et al., 2022). Some of these fields pertain to the characteristics of the publications. All ten fields are as follows:

- **Title**
- **Author(s)**
- **Year of publication**
- **Type of publication:** Journal, conference, or book chapter
- **Analysis type/objective:** Review classification, summarizing, prioritization, information extraction, information retrieval, visualization
- **Data and data sources:** App store(s), number of app categories, number of reviews
- **Analysis techniques**
- **Research outcome/result:** Tool or framework developed
- **Evaluation:** procedure, metrics, and criteria, result
- **Limitations and challenges of the study**

After organizing the data in the spreadsheet, classification and clustering methods were utilized to identify various review analysis types/categories and techniques. Additionally, previous literature related to prioritizing user concerns from app reviews was separately examined to identify specific techniques used, challenges, and limitations to address one of the objectives of this review. Finally, LLM-based research studies were critically analyzed to answer and address the last research question.

# RESULTS

Findings and their significance for each research question are presented in this section. Despite the extensive research in opinion mining and review analysis, this study focuses explicitly on app review analysis conducted using automated and/or semi-automated techniques over the past decade, excluding manual approaches. The selection criteria were designed to identify papers most relevant to techniques for prioritizing user concerns, resulting in the inclusion of 53 articles for review. Publications from 2014 to May 2024 were considered, and the distribution of studies per year is illustrated in Figure 2.

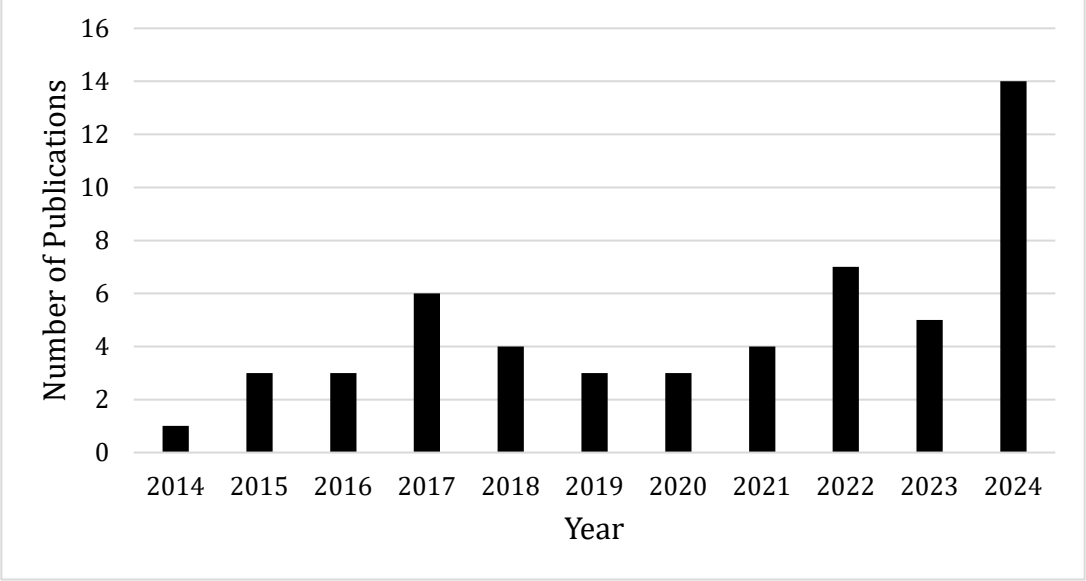


**Figure 2. Number of publications per year**

## *RQ1: CATEGORIES OF APP REVIEW ANALYSIS*

App review analysis has been conducted with various objectives and purposes. In this review, we have identified eight distinct categories of app review analysis, as illustrated in Figure 3. It is important to note that it is rarely seen for a study to employ only a single category of analysis. Furthermore, researchers have proposed and evaluated various tools and frameworks to automate app review analysis. Table 3 summarizes the tools that have been proposed by previous researchers to automate app review analysis, and the number of app reviews used to evaluate the tool.

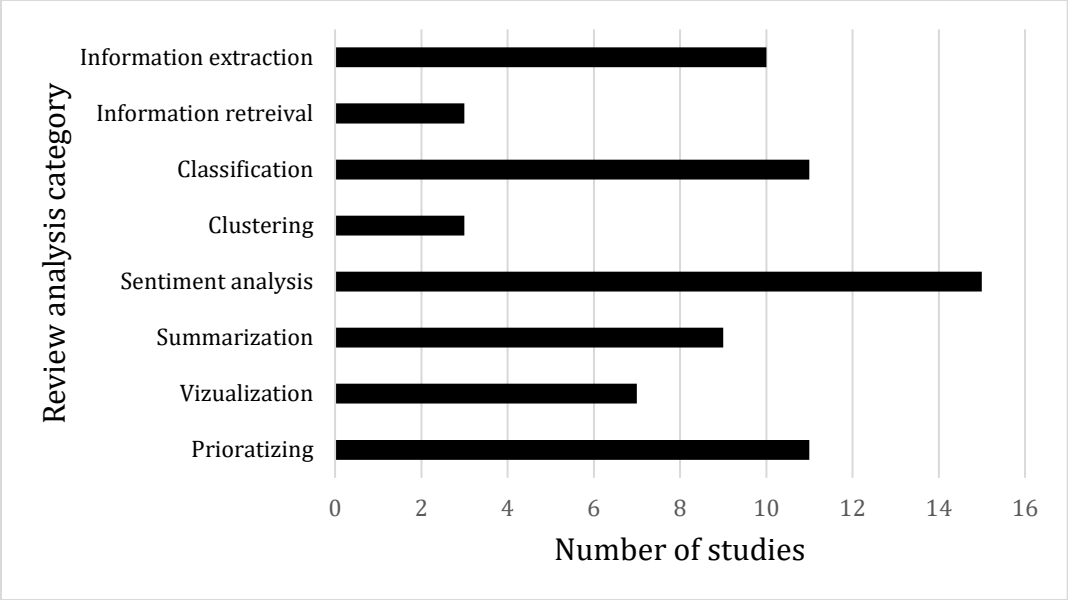


**Figure 3. Review analysis categories and their occurrences in the studied literature**

**Table 3. Tools and frameworks for app review analysis**

| Tool Frame-work | Purpose | Considered app stores/app categories and number of apps for evaluation | Number of reviews | Year | Reference |
|---|---|---|---|---|---|
| AR-Miner | Information extraction, clustering, prioritization & visualization | Play Store, 4 apps | - | 2014 | (N. Chen et al., 2014) |
| MERIT | Summarizing & visualization | Play Store (4 apps from 4 categories), App Store (2 apps from 2 categories) | 164,026 | 2015 | (Gao et al., 2022) |
| PAID | Information extraction, clustering, prioritization & visualization | Play Store (35 apps from 10 categories) | 2,089,737 | 2015 | (Gao et al., 2015) |
| MARK | Information extraction, classification & visualization | Play Store & App Store, 95 apps | - | 2015 | (Phong et al., 2015) |
| SURF | Summarizing | 17 apps | - | 2016 | (Di Sorbo et al., 2016) |
| URR | Classification & information retrieval | 39 apps | - | 2017 | (Ciurumelea et al., 2017) |
| IDEA | Information extraction, prioritization & visualization | Play Store (60 apps from 20 categories) (from the literature) | 164,026 | 2018 | (Gao et al., 2018) |
| MApp IDEA | Information extraction, prioritization & visualization | Play Store (60 apps from 20 categories) | 5 million | 2023 | (de Lima et al., 2023a) |

| Tool Frame-work | Purpose | Considered app stores/app categories and number of apps for evaluation | Number of reviews | Year | Reference |
|---|---|---|---|---|---|
| TOUR | Information extraction, summarization, & prioritization | | | | (T. Yang et al., 2021) |
| UX MAPPER | Summarizing | Play Store & App Store | - | 2024 | (W. T. Nakamura et al., 2024) |
| LLM-Cure | Classification, prioritization, information extraction, & information retrieval | 70 popular Android apps. | 1,056,739 | 2024 | (Assi et al., 2024) |
| SCRABLE | Information extraction, & information retrieval | 9 real-world customers | 49 | 2024 | (Azov et al., 2024) |
| LLM-based Risk Matrix | Information extraction, classification, & prioritization | 8 Popular apps (eBay, Evernote, Facebook, Netflix, Photo editor, Spotify, Twitter, Whatsapp ) | 363843 | 2023 | (de Lima et al., 2023b) |

## *RQ3: Techniques Used for Prioritizing User Concerns*

This section addresses RQ3 (What techniques are used for prioritizing user concerns?). The primary objective and significant focus of this literature review is to identify techniques employed to prioritize user concerns. Two main types of prioritizations are identified: review prioritization and user requirement prioritization (Malgaonkar et al., 2022). Table 4 summarizes the prioritization techniques used by previous researchers.

**Table 4. Summary of techniques used for prioritizing user concerns in app reviews**

| Category | Techniques | Purpose | References |
|---|---|---|---|
| **Review Prioritization** | Grouping-Based Ranking | Categorizing reviews into priority levels | (Gao et al., 2015) |
| | Regression Techniques (Time Series, Average Ratings) | Predicting review importance over time | (N. Chen et al., 2014) |
| | TOUR (Topic & Sentiment Analysis of User Reviews) | Dynamically prioritizing emerging issues based on app versions | (T. Yang et al., 2021) |
| **User Requirement Prioritization** | Anomaly Detection | Identifying unusual trends in reviews | (Gao et al., 2022) |
| | Risk Matrices (Clustering & Graph Theory) | Ranking concerns based on severity | (de Lima et al., 2023a) |
| | Thumbs-Up Count-Based Approach | Prioritizing reviews based on user upvotes | (Arambepola et al., 2024) |
| **LLM-Based Approaches** | OPT Model for Risk Matrices | Automating risk severity classification of user concerns | (de Lima et al., 2023b) |

### *RQ4: Challenges in app review analysis*

The summary of the critical challenges and common strategies used to mitigate them is shown in Table 5.

**Table 5. Challenges in app review analysis**

| Challenge | Strategies to overcome |
|---|---|
| Manual labelling | Multiple researchers are often involved in labeling, and statistical tests such as Cohen's Kappa are employed to assess inter-rater agreement and reduce bias (W. Nakamura et al., 2022). |
| Extracting meaningful informative reviews | Get the feedback from external professional app developers (Di Sorbo et al., 2016; Malgaonkar et al., 2022). |
| Evaluating the proposed approach and validating the results | Use official app change logs (Gao et al., 2022; C. Yang et al., 2021) and app descriptions (Y. Liu et al., 2018). |
| Generalizing the proposed framework | Validate with diverse app categories from different app stores. |
| Topic interpretation | Phase extraction over keyword extraction (Gao et al., 2015, 2022), mapping with the existing UX factors (Arambepola et al., 2024). |
| LLM context limitations | The batch-and-match method. This approach incrementally extracts the top *k* features from a large corpus of user reviews, ensuring efficient processing despite LLM context constraints (Assi et al., 2024). |
| Cost and availability of human evaluations | An automated system that can simulate human judgment in evaluating customer feedback, enabling real-time review assessments and fostering continuous service improvements (Azov et al., 2024). |
| Diverse descriptions and a large review volume for risk assessment | Use of automatic machine learning-based methods for extracting risks from reviews and classifying their priority (de Lima et al., 2023b). |

## DISCUSSION

The results presented in the previous section provide valuable insights into the mobile app user review analysis. This section presents the findings in the context of existing literature, highlighting their broader implications.

### *RQ1: Categories of App Review Analysis*

This section addresses RQ1 (What are the main categories of app review analysis in Software Engineering?). App review analysis has been conducted with various objectives and purposes. In this review, we have identified eight distinct categories of app review analysis, as illustrated in Figure 3. Most review analyses incorporate multiple categories to achieve their objectives (N. Chen et al., 2014; Gao et al., 2015; Phong et al., 2015). For example, AR-Miner is a computational framework consisting of components for extracting informative user reviews, clustering, prioritizing them using a review ranking schema, and visualization (N. Chen et al., 2014). Based on the prevalence in the selected literature, sentiment analysis is the most frequently used category, while review classification, information extraction, and information extraction are also widely employed in numerous studies. Furthermore, researchers have proposed and evaluated various tools and frameworks to automate app review analysis (Table 3). These tools support the different categories mentioned above to achieve their primary objectives. Reviews extracted from multiple apps across various app stores have been

utilized to validate the applicability, generalizability, and effectiveness of the proposed tools and frameworks.

Among the notable advancements in app review analysis, tools that involve LLM techniques such as LLM-Cure and SCRABLE (Assi et al., 2024; Azov et al., 2024) are crucial. LLM-Cure excels in accurately assigning features to user reviews and offers developers targeted suggestions for improving app features based on specific complaints by considering the positive reviews in competitive apps. SCRABLE focuses on customer review response generation by taking insights from real-world customers to craft meaningful and contextually relevant responses (Azov et al., 2024). These tools show the power of integrating LLMs in app review analysis. It enhances the overall quality of feedback interpretation and response generation by exploring the capabilities of LLMs, focusing on information retrieval and extraction at a more advanced level.

## *RQ2: Techniques used in App Review Analysis*

Answering RQ2 (Which techniques are employed for app review analyses?), even though manual analysis has been used in various domains, it is not feasible to carry out large-scale app review analysis due to its time-consuming nature, potential for bias, and human error. Consequently, this study excludes research papers that rely solely on manual analysis. Machine Learning (ML) and Natural Language Processing (NLP) have been the most widely used techniques for automating app review analysis. Additionally, various statistical methods are used to summarize and validate the proposed tools and approaches for app review analysis. Moreover, statistical techniques are also utilized for sampling selection and visualizing results through box plots and charts (W. Nakamura et al., 2022; Noei & Lyons, 2019).

ML techniques were primarily employed for the classification and clustering of app reviews. For instance, supervised ML techniques such as Naive Bayes, Support Vector Machine, and Random Forest are commonly used for binary and multiclass classification of reviews into categories such as 'bug reports,' 'end-user requests,' 'user experience,' and 'ratings' (Maalej et al., 2016; L. Wang et al., 2020). However, these techniques require labelled data as ground truth, which poses a limitation. Since user reviews are textual data, most studies utilize unsupervised ML techniques (de Lima et al., 2023a; Malgaonkar et al., 2022) or NLP techniques for review analysis. For example, various NLP techniques are applied for information extraction, summarization, and sentiment analysis. Among the NLP methods, topic modelling is extensively used, with Latent Dirichlet Allocation (LDA) being the most prevalent technique, and some studies adopting modified LDA approaches (Gao et al., 2018). Similarly, sentiment analysis has been implemented with various modifications (W. Nakamura et al., 2022). In addition, word embedding was utilized to identify synonyms for topic keywords in review summarizing and prioritization (Arambepola et al., 2024; Samy et al., 2021). Moreover, most research integrates multiple techniques to achieve its objectives in app review analysis. For example, sentiment analysis and LDA are commonly combined (Araujo et al., 2022).

Recent advancements have notably utilized LLMs such as Chat-GPT, Gemini, and LLaMA to enhance app review analysis (Gunathilaka & De Silva, 2022). These models have improved review classification, sentiment analysis, and feature extraction (Gunathilaka & De Silva, 2022; Roumeliotis et al., 2024). Additionally, these models can be further optimized by incorporating hyperparameters such as sequence length (maximum length of input tokens processed at once), temperature, and margin values (Assi et al., 2024). Also, recent developments in app review analysis leverage LLMs and RAG techniques to enhance the feature extraction process. One approach focuses on competitive user review analysis for feature enhancement, utilizing advanced LLMs, specifically the OPT model (de Lima et al., 2023b). This model's capabilities allow for a deeper understanding of user feedback by analyzing the nuances in user reviews, leading to more targeted feature improvements for applications. The integration of RAG techniques with LLMs enables researchers and developers to extract relevant features and generate insightful recommendations for enhancing app functionalities (Azov et al., 2024).

## *RQ3: Techniques Used for Prioritizing User Concerns*

This section addresses RQ3 (What techniques are used for prioritizing user concerns?). The primary objective and significant focus of this literature review is to identify techniques employed to prioritize user concerns. Two main types of prioritizations are identified: review prioritization and user requirement prioritization (Malgaonkar et al., 2022). Reviews are prioritized based on predefined criteria or metrics in review prioritization. In contrast, user requirement prioritization gives a list of prioritized user concerns and feedback by covering the overall user review dataset. Therefore, prioritizing the user requirement is worth identifying the features that need further improvements. Prioritization serves various objectives, including identifying emerging issues, optimizing release planning, and facilitating prompt feedback by minimizing the time between issue identification and resolution (de Lima et al., 2023b; Gao et al., 2015, 2018, 2022; Malgaonkar et al., 2022). Therefore, user requirement prioritization is particularly critical for enhancing the release planning of future app versions. Widely used prioritization techniques include anomaly detection methods (Gao et al., 2022), risk matrices combining clustering and graph theory approaches (de Lima et al., 2023a), thumbs-up count-based approach (Arambepola et al., 2024), grouping-based ranking methods (Gao et al., 2015), and regression techniques involving time series matrices and average ratings (N. Chen et al., 2014). Additionally, TOpic and sentiment analysis of User Reviews (TOUR) is a tool capable of dynamically identifying and prioritizing emerging issues based on the app version. The prioritization is based on the probability distribution of the reviews under the topic (C. Yang et al., 2021).

Recent advancements in user requirement prioritization have evolved by integrating LLMs, such as the OPT model, to automate risk matrix construction. Traditionally, constructing a risk matrix was manual and time-consuming, requiring stakeholders to sift through large volumes of reviews with varied descriptions. LLMs, specifically the OPT model, have automated this process by extracting relevant features and bugs from app reviews, classifying them by risk severity, and generating dynamic, customized risk matrices (de Lima et al., 2023b).

## *RQ4: Challenges in App Review Analysis*

Addressing RQ4 (What challenges are encountered in app review analysis?), data preparation (review/data preprocessing), framework/tool evaluation, and validation are the key stages where challenges arise. The summary of the critical challenges and common strategies used to mitigate them is shown in Table 5. For instance, manual labelling is inherently challenging due to biases and time constraints. Additionally, authors who are not professional app developers in academic settings may need help accurately categorizing reviews as informative or non-informative or assigning them to specific categories. This potential discrepancy can impact research findings. To address this, researchers often use multiple professional app developers as external validators to ensure the accuracy of categorization (Di Sorbo et al., 2016; Malgaonkar et al., 2022).

Furthermore, using official app changelogs as ground truth data is an effective strategy (Gao et al., 2022; T. Yang et al., 2021). App descriptions are also used as an evaluation approach in app review summarization (Y. Liu et al., 2018). Given the diversity of mobile apps available, generalizing app review analysis approaches across different applications can be particularly challenging. Researchers address this by leveraging data from multiple app stores and diverse app categories to validate their approaches (Al-Subaihin et al., 2021; Mcilroy et al., 2017). Moreover, after identifying topics by analyzing app reviews using methods like LDA, topic interpretation is challenging and requires manual effort (Arambepola et al., 2024). System Usability Scale (SUS) is also an established method for evaluating newly proposed frameworks or tools' usability, reliability, and efficiency (Hirave et al., 2019). There, the tool's usability is assessed through a test of the tool followed by a set of SUS questions on a five-point Likert scale of 'Strong Agreement' to 'Strong Disagreement'.

Additionally, several emerging challenges have surfaced in app review analysis. One significant challenge is the limitations of LLMs in processing extensive contexts, which can hinder scalability. The batch-and-match method has been introduced to address this challenge, allowing for the incremental

extraction of the top k features from a large corpus of user reviews, ensuring efficient processing despite LLM context constraints (Assi et al., 2024). Furthermore, the cost and availability of human evaluations can pose obstacles due to their limited availability and expense. To mitigate this, an automated system called LLM-as-a-Judge has been proposed, which simulates human judgment in evaluating customer feedback, enabling real-time review assessments and fostering continuous service improvements (Azov et al., 2024). Lastly, the diverse descriptions and large volume of reviews complicate risk assessment, necessitating the use of automatic machine learning-based methods to extract risks from reviews and classify their priority. These methods can effectively address the challenges posed by varied descriptions and substantial review volumes, enhancing the overall reliability of app review analysis (de Lima et al., 2023b).

## *RQ5: Utilizing LLMs in App Review Analysis*

This section addresses RQ5 (How LLMs have evolved for the app review analysis process?). LLMs have evolved significantly in the app review analysis process, enhancing various app review categories such as classification, sentiment analysis, information extraction, and information retrieval.

Initially, LLMs were used primarily for review classification and sentiment analysis, enabling automated systems to categorize feedback based on user sentiment and identify key concerns regarding key features (Morbidoni, 2023; Roumeliotis et al., 2024; Shah et al., 2024). State-of-the-art LLMs like GPT-4 and ChatGPT have demonstrated their ability to analyse app reviews regarding these classification and sentiment analysis processes (Roumeliotis et al., 2024; Shah et al., 2024). Over time, these LLMs have become more sophisticated with tools like LLMs-Cure, offering enhanced capabilities for analyzing feedback to identify underperforming features and suggest improvements based on user preferences (Assi et al., 2024), by showing the improvement in not only review classification and sentiment analysis but also information extraction and retrieval.

Recent advancements have utilized LLMs, specifically the OPT model, to automatically construct risk matrices from app reviews by extracting features and bugs mentioned in the reviews. This approach incorporates dynamic, automatic prompt generation to extract specific application characteristics, and it evaluates prompts that classify risks by severity, from negligible to critical. This facilitates standardized, automated risk assessment and matrix construction. Experimental results indicate that OPT competes well with proprietary models like GPT-3.5 in risk matrix generation, providing a significant step forward in software product maintenance and evolution (de Lima et al., 2023b). Further research has developed specialized prompt templates to enable ChatGPT's effectiveness in complex tasks, such as quadruple extraction, by employing a selection method on few-shot examples to fully leverage its in-context learning ability. Comparative evaluations against state-of-the-art models on public datasets reveal the capability boundaries of ChatGPT in such complex extraction tasks (Xu et al., 2023).

Challenges persist in tasks requiring complex information extraction, such as quadruple extraction (Xu et al., 2023). However, advancements like specialized prompt templates and few-shot learning are pushing the boundaries of LLM capabilities by improving prioritization, classifications, and information extraction in much more complex scenarios rather than identifying key features and sentiments (de Lima et al., 2023b). Overall, LLMs have become invaluable tools in app review analysis. With ongoing research, their ability to understand and process feedback will continue to improve, driving future user-centred and competitive app development. In the future, LLMs can play a transformative role by enabling more comprehensive, context-aware analysis through deep semantic understanding and automated extraction of app features. These advanced models, such as LLM-Cure, can enhance precision in feature-specific sentiment analysis and provide actionable insights that surpass traditional methods in areas such as information retrieval. Integrating LLMs into app review analysis also opens new opportunities for adaptive prompt engineering and RAG, which would facilitate real-time updates and targeted feature optimization with more user-centric application development.

The findings of this study extend beyond mobile app review analysis to broader domains such as customer feedback management, requirement engineering, UX research, and risk assessment. Automated techniques like sentiment analysis, topic modeling, and LLM-driven tools can be applied to various tasks. For example, help businesses to analyze user feedback in e-commerce (N. Chen et al., 2014; Guzman & Maalej, 2014), enhance software development by identifying key requirements (Maalej et al., 2016; L. Wang et al., 2020), and improve UX in chatbots and smart devices (Gunathilaka & De Silva, 2022). Additionally, prioritization frameworks used in app reviews can support risk assessment in FinTech and healthcare, ensuring compliance and service improvements (de Lima et al., 2023a; T. Yang et al., 2021). Future research can explore multi-modal data sources like voice and video reviews to enhance automated feedback analysis. Our findings align with prior studies highlighting sentiment analysis, topic modeling, and machine learning as dominant techniques in app review analysis. However, unlike earlier research, we emphasize the increasing role of LLMs in automating feature extraction, sentiment classification, and prioritization, particularly with models like OPT and ChatGPT. Additionally, while prior studies relied heavily on predefined heuristics and statistical techniques, our review highlights emerging approaches integrating LLMs and RAG techniques for dynamic, real-time prioritization. Furthermore, we identify challenges specific to LLM-based analysis, such as context limitations and evaluation constraints, which were less explored in previous literature.

While this systematic review covers a broad and concise range of techniques, several limitations exist. First, the review is limited to studies published within the last decade (2014–2024). While this time frame captures recent advancements in the field, it may not fully account for longer-term trends or foundational research that preceded this period. Second, the inclusion of only English-language publications may result in the exclusion of important studies published in other languages, limiting the diversity of techniques considered. Additionally, while grey literature was consulted to inform the statistical and contextual analysis, it was not included in the review due to concerns regarding the uncertainty and validity of such sources. However, this may limit the incorporation of the most recent developments or cutting-edge techniques that have not yet been formally published in peer-reviewed journals in this highly dynamic research field. Future research could refine LLM processing under resource constraints. One potential approach is the batch-and-match method, which incrementally extracts the top k features from extensive user reviews, optimizing processing within LLM context limits (Assi et al., 2024). Also, addressing ethical concerns in automated decision-making is crucial. Future studies should explore human-judgment-simulating systems for real-time customer feedback analysis, ensuring fairness and transparency (Azov et al., 2024). Moreover, automated machine learning methods for extracting and prioritizing risks from reviews can improve service quality while mitigating classification biases. Aligning these techniques with regulatory and ethical standards is essential for responsible AI deployment (de Lima et al., 2023b).

## CONCLUSION

This systematic review emphasizes the importance of app review analysis in Software Engineering, particularly for enhancing mobile app user experiences. The review examined studies published between 2014 and 2024 to identify critical categories, techniques, challenges, and strategies to overcome these challenges in app review analysis, including in novel LLM-based approaches. The results found that sentiment analysis, review classification, and information extraction are significant categories, with ML and NLP being the most commonly used techniques in app review analysis. Prioritizing user concerns is a crucial aspect of app review analysis. It has been achieved through anomaly detection, risk matrices, thumbs-up count-based approaches, and regression techniques. These methods help to optimize the release plan by identifying emerging issues from the user reviews. Difficulties in data preparation, biases in manual labelling for supervised machine learning, interpreting topics or themes identified from app reviews, and generalizing findings across diverse apps are the main challenges in review analysis. Moreover, LLM-based approaches, including ChatGPT and OPT, are emerging as

valuable tools for extracting meaningful insights from user reviews. In conclusion, systematic app review analysis is vital for improving app quality and user satisfaction by providing an optimized UX. In the future, app review analysis can be extended in multiple directions, such as incorporating large language models for enhanced sentiment analysis and topic modelling, conducting longitudinal studies to track and address persistent issues over time, and focusing on identifying ongoing user concerns to provide actionable insights for continuous app improvement.

# REFERENCES


Ahn, H., & Park, E. (2023). Motivations for user satisfaction of mobile fitness applications: An analysis of user experience based on online review comments. *Humanities and Social Sciences Communications*, *10*, Article 3. https://doi.org/10.1057/s41599-022-01452-6

Al-Subaihin, A. A., Sarro, F., Black, S., Capra, L., & Harman, M. (2021). App store effects on software engineering practices. *IEEE Transactions on Software Engineering*, *47*(2), 300–319. https://doi.org/10.1109/TSE.2019.2891715

Arambepola, N., Munasinghe, L., & Warnajith, N. (2024). Factors influencing mobile app user experience: An analysis of education app user reviews. *Proceedings of the 4th International Conference on Advanced Research in Computing, Belihuloya, Sri Lanka,* 223-228. https://doi.org/10.1109/icarc61713.2024.10499727

Araujo, A. F., Gôlo, M. P. S., & Marcacini, R. M. (2022). Opinion mining for app reviews: An analysis of textual representation and predictive models. *Automated Software Engineering*, *29*, Article 5. https://doi.org/10.1007/s10515-021-00301-1

Assi, M., Hassan, S., & Zou, Y. (2024). *LLM-Cure: LLM-based competitor user review analysis for feature enhancement.* PsyArXiv. https://doi.org/10.48550/arXiv.2409.15724

Azov, G., Pelc, T., Alon, A. F., & Kamhi, G. (2024). *Self-improving customer review response generation based on LLMs.* PsyArXiv. http://arxiv.org/abs/2405.03845

Boaye Belle, A., & Zhao, Y. (2022). Evidence-based software engineering: A checklist-based approach to assess the abstracts of reviews self-identifying as systematic reviews. *Applied Sciences*, *12*(18), 9017. https://doi.org/10.3390/app12189017

Chemnad, K., Alshakhsi, S., Almourad, M. B., Altuwairiqi, M., Phalp, K., & Ali, R. (2022). Smartphone usage before and during COVID-19: A comparative study based on objective recording of usage data. *Informatics*, *9*(4), 98. https://doi.org/10.3390/informatics9040098

Chen, N., Lin, J., Hoi, S. C. H., Xiao, X., & Zhang, B. (2014). AR-miner: Mining informative reviews for developers from mobile app marketplace. *Proceedings of the 36th International Conference on Software Engineering* (pp. 767–778). Association for Computing Machinery. https://doi.org/10.1145/2568225.2568263

Chen, Q., Chen, C., Hassan, S., Xing, Z., Xia, X., & Hassan, A. E. (2021). How should I improve the UI of my app? A study of user reviews of popular apps in the Google Play. *ACM Transactions on Software Engineering and Methodology*, *30*(3), Article 37. https://doi.org/10.1145/3447808

Chou, H.-M., & Cho, T.-L. (2023). Utilizing text mining for labeling training models from futures corpus in generative AI. *Applied Sciences*, *13*(17), 9622. https://doi.org/10.3390/app13179622

Ciurumelea, A., Schaufelbühl, A., Panichella, S., & Gall, H. C. (2017). Analyzing reviews and code of mobile apps for better release planning. *Proceedings of the IEEE 24th International Conference on Software Analysis, Evolution and Reengineering, Klagenfurt, Austria,* 91–102. https://doi.org/10.1109/SANER.2017.7884612

Dąbrowski, J., Letier, E., Perini, A., & Susi, A. (2022). Mining user feedback for software engineering: Use cases and reference architecture. *Proceedings of the IEEE International Conference on Requirements Engineering, Melbourne, Australia*, 114–126. https://doi.org/10.1109/RE54965.2022.00017

Dąbrowski, J., Letier, E., Perini, A., & Susi, A. (2023). Mining and searching app reviews for requirements engineering: Evaluation and replication studies. *Information Systems*, *114*, 102181. https://doi.org/10.1016/j.is.2023.102181

de Lima, V. M. A., Barbosa, J. R., & Marcacini, R. M. (2023a). *Issue detection and prioritization based on app reviews*. https://doi.org/10.21203/rs.3.rs-2838568/v1

de Lima, V. M. A., Barbosa, J. R., & Marcacini, R. M. (2023b). *Learning risk factors from app reviews: A large language model approach for risk matrix construction*. https://doi.org/10.21203/rs.3.rs-3182322/v1

Dennstädt, F., Zink, J., Putora, P. M., Hastings, J., & Cihoric, N. (2024). Title and abstract screening for literature reviews using large language models: An exploratory study in the biomedical domain. *Systematic Reviews*, *13*, Article 158. https://doi.org/10.1186/s13643-024-02575-4

Di Sorbo, A., Panichella, S., Alexandru, C. V., Shimagaki, J., Visaggio, C. A., Canfora, G., & Gall, H. C. (2016). What would users change in my app? summarizing app reviews for recommending software changes. *Proceedings of the 24th ACM SIGSOFT International Symposium on Foundations of Software Engineering* (pp. 499–510). Association for Computing Machinery. https://doi.org/10.1145/2950290.2950299

Gao, C., Wang, B., He, P., Zhu, J., Zhou, Y., & Lyu, M. R. (2015, November). PAID: Prioritizing app issues for developers by tracking user reviews over versions. *Proceedings of the IEEE 26th International Symposium on Software Reliability Engineering, Gaithersbury, MD, USA,* 35–45. https://doi.org/10.1109/ISSRE.2015.7381797

Gao, C., Zeng, J., Lyu, M. R., & King, I. (2018). Online app review analysis for identifying emerging issues. *Proceedings of the 40th International Conference on Software Engineering* (pp. 48–58). Association for Computing Machinery. https://doi.org/10.1145/3180155.3180218

Gao, C., Zeng, J., Wen, Z., Lo, D., Xia, X., King, I., & Lyu, M. R. (2022). Emerging app issue identification via online joint sentiment-topic tracing. *IEEE Transactions on Software Engineering*, *48*(8), 3025–3043. https://doi.org/10.1109/tse.2021.3076179

Genc-Nayebi, N., & Abran, A. (2017). A systematic literature review: Opinion mining studies from mobile app store user reviews. *Journal of Systems and Software*, *125*, 207–219. https://doi.org/10.1016/j.jss.2016.11.027

Gunathilaka, S., & De Silva, N. (2022, November). Aspect-based sentiment analysis on mobile application reviews. *Proceedings of the 22nd International Conference on Advances in ICT for Emerging Regions*, *Colombo, Sri Lanka,* 183–188. https://doi.org/10.1109/ICTer58063.2022.10024070

Guzman, E., & Maalej, W. (2014, August 1). How do users like this feature? A fine grained sentiment analysis of app reviews. *IEEE 22nd International Requirements Engineering Conference (RE)*, Karlskrona, Sweden, 2014, pp. 153-162, https://doi.org/10.1109/RE.2014.6912257

Haggag, O., Grundy, J., Abdelrazek, M., & Haggag, S. (2022). A large scale analysis of mHealth app user reviews. *Empirical Software Engineering*, *27*, Article 196. https://doi.org/10.1007/s10664-022-10222-6

Hirave, T., Malgaonkar, S., Alwash, M., Cheriyan, J., & Surve, S. (2019, December). Analysis and prioritization of app reviews. *Proceedings of the International Conference on Advances in Computing, Communication and Control, Mumbai, India,* 1–8. https://doi.org/10.1109/ICAC347590.2019.9036801

Jeong, N., & Lee, J. (2024). An aspect-based review analysis using ChatGPT for the exploration of hotel service failures. *Sustainability*, *16*(4), 1640. https://doi.org/10.3390/su16041640

Kitchenham, B., & Charters, S. (2007). *Guidelines for performing systematic literature reviews in software engineering*. Technical Report EBSE 2007-001. https://legacyfileshare.elsevier.com/promis_misc/525444systematicreviewsguide.pdf

Kozlowski, D., Pradier, C., & Benz, P. (2024). *Generative AI for automatic topic labelling*. PsyArXiv. https://arxiv.org/abs/2408.07003

Li, Y., Jia, B., Guo, Y., & Chen, X. (2017). Mining user reviews for mobile app comparisons. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, *1*(3), 1–15. https://doi.org/10.1145/3130935

Lim, Z. Y., Ong, L. Y., & Leow, M. C. (2021). A review on clustering techniques: Creating better user experience for online roadshow. *Future Internet*, *13*(9). https://doi.org/10.3390/fi13090233

Liu, Y., Liu, L., Liu, H., & Wang, X. (2018). Analyzing reviews guided by App descriptions for the software development and evolution. In *Journal of Software: Evolution and Process* (Vol. 30, Issue 12). John Wiley and Sons. https://doi.org/10.1002/smr.2112

Liu, Z., Chen, C., Wang, J., Chen, M., Wu, B., Tian, Z., Huang, Y., Hu, J., & Wang, Q. (2024). Testing the limits: Unusual text inputs generation for mobile app crash detection with large language model. *Proceedings - International Conference on Software Engineering*, 1685–1696. https://doi.org/10.1145/3597503.3639118

Luiz, W., Viegas, F., Alencar, R., Mourão, F., Salles, T., Carvalho, D., Gonçalves, M. A., & Rocha, L. (2018, April). A feature-oriented sentiment rating for mobile app reviews. *Proceedings of the 2018 World Wide Web Conference*, *Lyon, France*, 1909–1918. https://doi.org/10.1145/3178876.3186168

Maalej, W., Kurtanović, Z., Nabil, H., & Stanik, C. (2016). On the automatic classification of app reviews. *Requirements Engineering*, *21*(3), 311–331. https://doi.org/10.1007/s00766-016-0251-9

Malgaonkar, S., Licorish, S. A., & Savarimuthu, B. T. R. (2022). Prioritizing user concerns in app reviews – A study of requests for new features, enhancements and bug fixes. *Information and Software Technology*, *144*, 106798. https://doi.org/10.1016/j.infsof.2021.106798

Mcilroy, S., Shang, W., Ali, N., & Hassan, A. E. (2017). User reviews of top mobile apps in Apple and Google app stores. *Communications of the ACM*, *60*(11), 62–67. https://doi.org/10.1145/3141771

Morbidoni, C. (2023). Poster: LLMs for online customer reviews analysis: Oracles or tools? Experiments with GPT 3.5. *Proceedings of the 15th Biannual Conference of the Italian SIGCHI Chapter* (Article 38). Association for Computing Machinery. https://doi.org/10.1145/3605390.3610810

Nakamura, W., Oliveira, E., de Oliveira, E., Redmiles, D., & Conte, T. (2022). What factors affect the UX in mobile apps? A systematic mapping study on the analysis of app store reviews. *Journal of Systems and Software*, *193*, 111462. https://doi.org/10.1016/j.jss.2022.111462

Nakamura, W. T., de Oliveira, E. C., de Oliveira, E., & Conte, T. (2024). UX-MAPPER: A user eXperience method to analyze app store reviews. *Proceedings of the XXII Brazilian Symposium on Human Factors in Computing Systems*. Association for Computing Machinery. https://doi.org/10.1145/3638067.3638109

Noei, E., & Lyons, K. (2019). A survey of utilizing user-reviews posted on Google play store. *Proceedings of the 29th Annual International Conference on Computer Science and Software Engineering* (pp. 54–63). IBM.

Page, M. J., McKenzie, J. E., Bossuyt, P. M., Boutron, I., Hoffmann, T. C., Mulrow, C. D., Shamseer, L., Tetzlaff, J. M., Akl, E. A., Brennan, S. E., Chou, R., Glanville, J., Grimshaw, J. M., Hróbjartsson, A., Lalu, M. M., Li, T., Loder, E. W., Mayo-Wilson, E., McDonald, S., … Moher, D. (2021). The PRISMA 2020 statement: An updated guideline for reporting systematic reviews. *BMJ*, 372(71). https://doi.org/10.1136/bmj.n71

Phong, M. V., Nguyen, T. T., Pham, H. V., & Nguyen, T. T. (2015). Mining user opinions in mobile app reviews: A keyword-based approach (T). *Proceedings of the 30th IEEE/ACM International Conference on Automated Software Engineering, Lincoln, NE, USA,* 749–759. https://doi.org/10.1109/ASE.2015.85

Rodrigues, P., Silva, I., Barbosa, G., Coutinho, F., & Mourão, F. (2017). Beyond the stars: Towards a novel sentiment rating to evaluate applications in web stores of mobile apps. *Proceedings of the 26th International Conference on World Wide Web Companion* (pp. 109–117). International World Wide Web Conferences Steering Committee. https://doi.org/10.1145/3041021.3054139

Roumeliotis, K. I., Tselikas, N. D., & Nasiopoulos, D. K. (2024). LLMs in e-commerce: A comparative analysis of GPT and LLaMA models in product review evaluation. *Natural Language Processing Journal*, *6*, 100056. https://doi.org/10.1016/j.nlp.2024.100056

Sami, A. M., Rasheed, Z., Kemell, K.-K., Waseem, M., Kilamo, T., Saari, M., Duc, A. N., Systä, K., & Abrahamsson, P. (2024). *System for systematic literature review using multiple AI agents: Concept and an empirical evaluation.* PsyArXiv. http://arxiv.org/abs/2403.08399

Samy, H., Helmy, A., & Ramadan, N. (2021). Aspect-based sentiment analysis of mobile apps reviews using class association rules and LDA. *Proceedings of the Tenth International Conference on Intelligent Computing and Information Systems*, *Cairo, Egypt,* 183–189. https://doi.org/10.1109/ICICIS52592.2021.9694242

Shah, F. A., Sabir, A., & Sharma, R. (2024). *A fine-grained sentiment analysis of app reviews using large language models: An evaluation study*. PsyArXiv. http://arxiv.org/abs/2409.07162

Srisopha, K., Phonsom, C., Li, M., Link, D., & Boehm, B. (2020). On building an automatic identification of country-specific feature requests in mobile app reviews: Possibilities and challenges. *Proceedings of the*

*IEEE/ACM 42nd International Conference on Software Engineering Workshops* (pp. 494–498). Association for Computing Machinery. https://doi.org/10.1145/3387940.3391492

Sultana, R., & Sarker, S. (2018). App review mining and summarization. *International Journal of Computer Applications*, *179*(38), 45–52. https://doi.org/10.5120/ijca2018916918

Upadhye, A. (2024). Sentiment analysis using large language models: Methodologies, applications, and challenges. *International Journal of Computer Applications*, *186*(20), 30–34. https://doi.org/10.5120/ijca2024923625

Villarroel, L., Bavota, G., Russo, B., Oliveto, R., & Di Penta, M. (2016). Release planning of mobile apps based on user reviews. *Proceedings of the 38th International Conference on Software Engineering* (pp. 14–24). Association for Computing Machinery. https://doi.org/10.1145/2884781.2884818

Wang, L., Nakagawa, H., & Tsuchiya, T. (2020). Opinion Analysis and Organization of Mobile Application User Reviews. *REFSQ Workshops*.

Wang, Z., Zhu, Y., & Zhang, Q. (2024). LLM for sentiment analysis in e-commerce: A deep dive into customer feedback. *Applied Science and Engineering Journal for Advanced Research*, *3*(4), 8–13. https://doi.org/10.5281/zenodo.12730477

Xu, X., Zhang, J.-D., Xiao, R., & Xiong, L. (2023). *The limits of ChatGPT in extracting aspect-category-opinion-sentiment quadruples: A comparative analysis.* PsyArXiv. https://doi.org/10.48550/arXiv.2310.06502

Yang, C., Wu, L., Yu, C., & Zhou, Y. (2021). A phrase-level user requests mining approach in mobile application reviews: Concept, framework, and operation. *Information*, *12*(5), 177. https://doi.org/10.3390/info12050177

Yang, T., Gao, C., Zang, J., Lo, D., & Lyu, M. (2021). TOUR: Dynamic topic and sentiment analysis of user reviews for assisting app release. *Companion Proceedings of the Web Conference* (pp. 708–712). Association for Computing Machinery. https://doi.org/10.1145/3442442.3458612

Yu, D., Mu, Y., & Jin, Y. (2017). Rating prediction using review texts with underlying sentiments. *Information Processing Letters*, *117*, 10–18. https://doi.org/10.1016/j.ipl.2016.08.002

Zhang, W., Deng, Y., Liu, B., Pan, S. J., & Bing, L. (2024). Sentiment analysis in the era of large language models: A reality check. *Findings of the Association for Computational Linguistics* (pp. 3881–3906). Association for Computational Linguistics. https://doi.org/10.18653/v1/2024.findings-naacl.246

Zhao, Y., Hou, X., Wang, S., & Wang, H. (2024). LLM app store analysis: A vision and roadmap. *ACM Transactions on Software Engineering and Methodology*. https://doi.org/10.1145/3708530

## AUTHORS

**Nimasha Arambepola** is a Probationary Lecturer at the Software Engineering Teaching Unit, Faculty of Science, University of Kelaniya, Sri Lanka. She earned her BSc (Hons) in Software Engineering with First Class Honours in 2020 from the same university, and is currently pursuing an MPhil in Software Engineering. Her research interests include Software Engineering, Human-Computer Interaction (HCI), User Factor Analysis, Information Extraction, and Natural Language Processing.

**Dr Lankeshwara Munasinghe** is a Lecturer in Cyber Security at the School of Computing, Engineering, and Technology, Robert Gordon University, with a strong background in cybersecurity, AI security, and research software engineering. His research focuses on cybersecurity knowledge graphs, AI-driven security solutions, and social network security, particularly in misinformation spread and anomaly detection. Dr. Munasinghe holds a PhD in Informatics from The Graduate University for Advanced Studies (SOKENDAI), Japan, and a BSc (Hons) in Statistics & Computing from the University of Kelaniya. His professional experience includes roles as a Senior Lecturer at the University of Kelaniya, a Visiting Researcher at Hokkaido University, and a Senior Staff Engineer at Motorola Solutions, where he contributed to human activity recognition systems for security applications.

**R. A. W. L. Wimalasena** is a Temporary Demonstrator at the Software Engineering Teaching Unit, University of Kelaniya, Sri Lanka. She holds a First Class BSc (Hons) degree in Software Engineering from the same university. Her research interests focus on Artificial Intelligence, particularly Deep Learning and Generative AI, and sustainable web development technologies. She has contributed to academic publications and remains actively engaged in research within her areas of expertise.