Research Article

An Observational Study on React Native (RN) Questions on Stack Overflow (SO)

Luluh Albesher (iD)**, Razan Aldossari, and Reem Alfayez** (iD)

College of Computer and Information Sciences, King Saud University, Riyadh, Saudi Arabia

Correspondence should be addressed to Reem Alfayez; reem.alfayez@hotmail.com

Mobile applications are continuously increasing in prevalence. One of the main challenges in mobile application development is creating cross-platform applications. To facilitate developing cross-platform applications, the software engineering community created several solutions, one of which is React Native (RN), which is a popular cross-platform framework. The software engineering literature demonstrated the effectiveness of Stack Overflow (SO) in providing real-world perspectives on a variety of technical subjects. Therefore, this study aims to gain a better understanding of the stance of RN on SO. We identified and analyzed 131,620 SO RN-related questions. Moreover, we observed how the interest toward RN on SO evolves over time. Additionally, we utilized Latent Dirichlet Allocation (LDA) to identify RN-related topics that are discussed within the questions. Afterward, we utilized a number of proxy measures to estimate the popularity and difficulty of these topics. The results revealed that interest toward RN on SO was generally increasing. Moreover, RN-related questions revolve around six topics, with the topics of layout and navigation being the most popular and the topic of iOS issues being the most difficult. Software engineering researchers, practitioners, educators, and RN contributors may find the results of this study beneficial in guiding their future RN efforts.

1. Introduction

The demand and usage of mobile applications has been continuously growing over time [1–3]. Mobile applications are becoming ubiquitous that pertain to many aspects of life, such as education [4], health [5], finance [6], transportation [7], entertainment [3], and more [3]. The number of mobile applications that can be downloaded on Google Play Store (https://play.google.com/store) and Apple App Store (https://apps.apple.com/) combined is above 5 million applications. A report on the usage of mobile applications divulged that the revenue of Google Play Store and Apple App Store in the third quarter of 2021 is over 30 billion US dollars. The revenue of mobile applications is expected to increase in the future, with a prediction to reach more than 613 billion US dollars in 2025 [8].

Though the reliance on mobile applications is growing in popularity, with tremendous potential and opportunities at hand, mobile application development is not an easy feat [2, 9–11]. Mobile application development is considered to be distinct from traditional software system development,

and it has its unique set of challenges [1, 2, 12–15]. One of the core challenges that resides within mobile application development is the development of cross-platform applications. A cross-platform application refers to a mobile application that runs seamlessly on multiple operating systems, such as Android and iOS [2, 11]. As mobile applications are expected to be compatible with multiple platforms, the software engineering community has gone to great lengths to aid and facilitate the process of developing cross-platform applications by creating multiple solutions, with an aim to decrease the cost of, speed, and simplify developing cross-platform applications [15–17]. Particularly, to overcome challenges within the development of cross-platform applications, the software engineering community produced several solutions, such as practices, tools, and technologies [2, 17–20]. One of these solutions is cross-platform development frameworks, which allow mobile developers to develop one codebase that can be run on multiple platforms, aiming to produce cross-platform applications more efficiently and effectively [21, 22].

React Native (RN) (https://reactnative.dev/) is an open-source cross-platform development framework created by

Meta, known previously as Facebook [21, 22]. RN was developed and used internally by Meta in 2013, and it was officially released in 2015 [23, 24]. The framework enables developers to build a single application that can be run on multiple platforms, such as iOS and Android, through native rendering for each platform [25]. RN is a popular mobile development framework that has been utilized to build thousands of mobile applications, such as Skype, Microsoft Teams, Shopify, Tableau, and more [21, 26]. Furthermore, Stack Overflow (SO) conducts an annual survey that aims to understand trends within the technology field. In the 2022 SO survey (https://survey.stackoverflow.co/2022/) in which over 70,000 developers from over 180 countries took part, not only RN was found to be one of the most three popular cross-platform frameworks but also one of the top three most wanted to learn.

Upon examining the software engineering literature, we found that SO has been successfully utilized to gain a better understanding of various subjects in the software engineering field [2, 27–31], as the website is considered to be the largest, most popular, and most trusted technical online Q&A website [2, 28]. Specifically, analyzing questions related to a specific subject was found to be fruitful in understanding the interest toward the subject on the website, identifying topics that are discussed within the subject, and estimating the popularity and difficulty of these identified topics [2, 28, 30].

Acquiring a better understanding on the stance of RN on SO can be beneficial from multiple perspectives. RN practitioners can gain insights into the challenges they might encounter while working with RN and, by extension, within the broader field of mobile development. Both RN contributors and software engineering researchers can identify areas in which RN and mobile development can be enhanced. Lastly, by unveiling and understanding the challenges within the usage of RN, software engineering educators can tailor their teaching curricula to address these challenges effectively. Therefore, in this study, we conducted an empirical analysis that is focused on RN-related questions on SO. The study was initiated by identifying and acquiring 131,620 RN-related questions. Afterward, we analyzed these questions to gauge how the interest toward RN on SO evolved over time. We then utilized Latent Dirichlet Allocation (LDA), a topic modeling technique, to understand what RN-related topics are discussed on the website [32]. Then, we leveraged a number of proxy measures to estimate the popularity and difficulty of each identified topic. The main contributions of this study can be summarized as follows:

(1) Identifying and analyzing a set of 131,620 SO RN-related questions.

(2) Revealing that the interest toward RN on SO is generally increasing.

(3) Identifying the six main topics that SO users are asking about with respect to RN.

(4) Assessing the popularity and difficulty of these identified six topics.

The rest of the paper is organized as follows: background information is presented in Section 2; an overview of previous, related work is summarized in Section 3; the setup of the study is described in Section 4; Section 5 presents the results of the study, and Section 6 discusses the results; Section 7 summarizes implications of the study; Section 8 presents threats to the validity of the study and their corresponding mitigation strategies, once applicable; lastly, the paper is concluded in Section 9.

2. Background

This section presents background information that aids in understanding the study. Specifically, we provide a summarized overview of the Stack Exchange network and SO, RN, and the LDA algorithm. We note that the descriptions of the Stack Exchange network and SO are derived from the Stack Exchange network website meta, and the summary of RN is sourced from its official documentation.

2.1. Stack Exchange Network and Stack Overflow (SO). Stack Exchange network contains over 170 Q&A websites that span over multiple fields. The main idea of the network is helping users gaining and sharing knowledge through asking questions, answering questions, or commenting on questions and answers. Questions are required to be labeled with at least one tag and can have up to five tags. A tag is a keyword that aids in categorizing and searching questions. Users can also upvote or downvote questions based on how they perceive the value of a question. Websites on the network are self-moderated through a reputation system. This system rewards users based on the quality of their contributions (i.e., questions, comments, answers, and edits), where a user's reputation is increased when other users upvote their contributions, and it decreases if the user's contributions are downvoted.

The largest, oldest, and most popular Stack Exchange community website is SO. Since the website's inception in 2008, the number of its users has increased, and the website is estimated to have more than 7 million daily visits and around 5,000 daily posted questions. As of December 07, 2022, SO has 19,330,674 million users, 23,147,406 million questions, 34,321,516 million answers, and 87,107,650 million comments. The website caters to both amateur and professional users, and it serves as a platform in which they can seek and give answers to technical issues. Questions on the website cover a wide range of coding and programing issues, such as a specific software development problem, encountered bug, algorithm, or a testing problem.

2.2. React Native (RN). RN is an open-source framework for building cross-platform applications using JavaScript based on React [23]. The framework was developed and utilized internally by Meta since 2013, and it was publicly released in 2015 [23]. Since the framework release, it has proven its capability in creating robust and responsive mobile applications effectively and efficiently [21]. Not only the framework has been adopted by many users, but many major companies

```
React component
render: function() {
    return <View> Hello
World! </View> ;
}
```
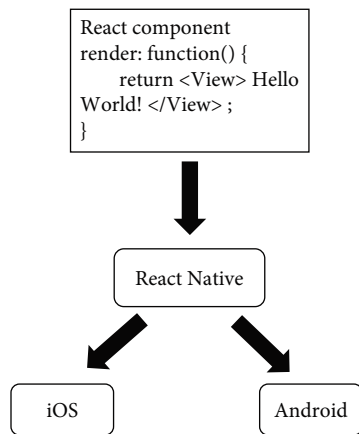
React Native

iOS   Android

FIGURE 1: An illustration of bridging in RN.

have also invested heavily in the framework, such as Microsoft, Discord, and Shopify [26].

The framework grants its users the ability to write code once and deploy it across multiple platforms. As Figure 1 depicts, this is accomplished through a process known as bridging, which refers to the process of connecting JavaScript code to the native code of a platform [21]. This enables developers to write a single codebase while still having access to native functionalities and components that are specific to each platform. When an RN application is compiled, the JavaScript code is bundled and shipped with the application, and at runtime, the JavaScript code communicates with the native code through the bridge [21]. The bridge allows the JavaScript code to interact with native components such as the camera and push notifications [21].

In RN, components are the building blocks of an application [25]. A component is a reusable piece of code that defines a part of the user interface that can also be combined to create more complex user interfaces. The framework provides a range of standard (i.e., built-in) components that developers can utilize across platforms, such as Buttons, Text, View, and Input. These components are designed to look and behave the same way on each platform, providing a consistent user experience. Additionally, developers can create custom components that can be reused across different parts of an application or even across different applications [22]. The appearance of a component can be customized through properties, which are values that are passed to a component when it is created. The behavior or state of a component can be altered through methods, which are functions that can be used to update the component's behavior or state. Components can also be further categorized into class or function components, where function components used to be stateless until a recent update that enabled function components to have state [33].

One of the key benefits of the framework is that it enables developers to create high-performance native mobile applications using a familiar web development toolset, which makes it easier for web developers to transit to mobile application development and allows them to leverage their existing skills and knowledge [22, 34, 35]. Another benefit is that

the framework can lower the cost of application development and increase the productivity of developers while preserving a native look and feel for the developed applications [22].

2.3. Latent Dirichlet Allocation (LDA). Topic modeling is an unsupervised machine-learning technique that attempts to derive main topics from a huge collection of text. LDA is a topic modeling algorithm that has been widely utilized in the field of software engineering to gain an insight into developers' discussions that revolve around a variety of subjects due to the algorithm's effectiveness and efficiency [28, 30, 36–38]. In the context of SO, the input to the algorithm is a set of SO questions. The algorithm groups words extracted from the set of questions based on the frequency of their cooccurrence appearance. Subsequently, the algorithm assigns a probability score to each question that indicates the question's relation to each extracted topic. Deciding the topic to which a question belongs to is based on the obtained probability scores; questions are categorized under topics that scored the highest [2, 28]. Moreover, the algorithm outputs the top-10 keywords that appear the most in each extracted topic [2, 28, 30, 36].

One challenge within the usage of the algorithm is determining the optimal number of topics, as the number of topics is one of the required inputs to the algorithm. Determining the optimal number of topics is particularly crucial, as the quality of the obtained results is highly influenced by the number of topics [2, 32, 36]. Assigning a large number of topics may lead to coarse-grained topics, which may complicate the analysis of topics and result in useless topics [2, 32, 36]. On the other hand, assigning a small number of topics may lead to overlapping topics that are challenging to analyze [2, 32, 36]. This challenge can be overcome by conducting multiple experiments with varying number of topics [2, 30, 36, 37, 39]. The initial number of topics can be set to two, and it should be incremented in each experiment run by a predefined increment. Additionally, in each experiment run, the coherence score, which is a model quality indicator, must be recorded. A higher coherence score indicates better model quality, though it is not an absolute guarantee. Once the experiment runs are concluded, the model that obtained the highest coherence score can be considered the optimal model, and therefore, its number of topics can also be utilized as the optimal number of topics [2, 30, 40].

Another challenge that resides within the utilization of the algorithm is that the algorithm only groups questions into topics without naming these topics [28]. Fortunately, one can follow a manual approach to name these topics [28], such as open card sort [41] or thematic analysis [42].

3. Related Work

SO has been utilized by multiple previous, related works to gain a better understanding of mobile development. Many studies have aimed to understand mobile-development-related questions in general, while other studies have either focused on specific mobile platforms or programing

languages. To the best of our knowledge, there is no previous study that is focused on RN.

Linares-Vásquez et al. [10] analyzed over 400,000 SO mobile-related questions. The authors applied LDA to understand the main discussed topics. Then, they analyzed the questions to gain an insight into the answering trends. The analysis revealed that SO users focus their answering on one platform, such as only answering Android-related questions. In addition, the results revealed that some of the identified topics have more accepted answers, including the topics of data types, compatibility, and layout. Conversely, questions related to accessing/browsing web content and graphics editing topics were found to have lower rates of accepted answers. Similarly, our study found that the layout topic has a relatively high accepted answer rate in comparison to the other identified RN-related topics. Similarly, Rosen and Shihab [2] analyzed over 13 million mobile-related posts on SO to gain an insight into mobile-related topics that SO users ask about. The authors utilized LDA on the retrieved posts, and they found that the identified set of questions can be categorized into application distribution, user interface development, mobile tools, mobile APIs, sensors and context, and data management. The study found that application distribution, user interface development, and mobile tools were the most popular categories of topics. Both Rosen and Shihab's study and our study found that topics that are relevant to the mobile application look (i.e., user interface and layout) are among the most popular mobile-related topics. Furthermore, Rosen and Shihab [2] found that mobile APIs was one of the most difficult category of topics. The authors furthered their analysis by exploring platform-specific issues and found that different mobile platforms have different issues; Android developers were mainly struggling with the open API of their platform, while iOS developers were mainly asking about distributing and deploying applications. Lastly, the study explored what type of questions are asked by SO users with regard to mobile development and found that questions are mainly focused on seeking instructions, which may indicate an increased need for better documentation.

Beyer and Pinzger [43] conducted a manual analysis of 450 SO Android-related questions. The result revealed that the majority of these questions were of the types of "How to" and "What is the problem." Additionally, the study found that questions related to user interface and core elements were the most frequently asked topics. Similarly, our study found that the layout and navigation topic to be the most discussed and the most popular RN-related topic. Oliveira et al. [44] analyzed over 9,000 Kotlin-related questions retrieved from SO using LDA and expanded the scope of their exploration by conducting semi-structured interviews with seven practitioners, which are mobile developers, mobile engineers, Android test analysts, mobile leader engineers, and mobile consultants. The study found that SO Kotlin-related discussions revolve around seven categories of topics. The study also revealed that though developers perceive Kotlin to be easy to use and have the potential in improving code quality, developers face many challenges when utilizing the language, such as interoperating with Java.

Several other works have utilized SO and LDA to gain an insight into multiple software engineering subjects, such as security [45], concurrency [29], ethics [36], microservices [46], big data [47], chatbot [31], and more [48, 49]. Moreover, Silva et al. [28] surveyed the literature of topic modeling in software engineering research, and we recommend the reader to reference the survey for a more comprehensive overview on topic modeling in software engineering.

4. Study Setup

This section outlines the setup of the study: research questions and utilized dataset.

4.1. Research Questions. This study addresses three research questions that were derived to attain the goal of this study: acquiring a better understanding of the stance of RN on SO.

4.1.1. Interest. As this study's primary objective is to gain a better understanding of the stance of RN on SO, it is only imperative to gauge the interest toward the framework on the website and how it evolved over time. Hence, the first research question of this study is: *RQ1: How does the interest toward RN on SO evolve over time?*

4.1.2. Topics. The interest toward mobile applications and their development has been steadily growing [2, 3]. Mobile application development is considered to be unique from traditional software development, and it has a distinguished set of challenges [1, 2, 12–15]. Despite RN's aim to ease the process of mobile application development, the utilization of the framework still demands overcoming some of the innate hurdles within the development of mobile application development, such as optimizing layout and power consumption [2, 15]. Thus, RN developers may encounter many challenges when using the frameworks. As this study's main aim is to acquire a better understanding of the stance of RN on SO, it is crucial to understand RN-related topics that are discussed by the website users. Therefore, the second research question of this study is: *RQ2: What RN-related topics are discussed on SO?*

4.1.3. Topic Characteristics. Though the second research question aims to identify the topics related to RN that are discussed on SO, the question fails to provide any information about how popular or difficult these topics are. Estimating the popularity of these topics can be beneficial in determining which topics are of most interest to developers [2, 30, 36, 37, 50]. Moreover, identifying which RN-related topics are more difficult can be beneficial in determining which topics are considered more important and require more attention [2, 30, 36, 37, 50]. Furthermore, gaining an insight regarding the monotonic correlation between popularity and difficulty grants a comprehensive understanding of the relationship between popularity and difficulty [2, 30, 37, 50]. This leads to the study's third research question being: *RQ3: What are the characteristics of the identified topics, in terms of popularity and difficulty, and how are popularity and difficulty related?*

4.2. Dataset. The first step to answer this study's research questions was obtaining a dataset that contains SO RN-related

questions. To acquire these questions, we utilized the Stack Exchange Data Dump (https://archive.org/details/stackexchange), which is a collection of XML files containing anonymized data of all user contributions on Stack Exchange websites. The data used in this study were the SO Stack Exchange Data Dump, including data until December 07, 2022. To identify RN-related questions in the utilized dump, we applied a search method that includes both tag-based and content-based filterings, as utilized by other studies [30, 36, 49]. The method involves searching for questions in the dataset that contains the term "react_native" where "_" represents any character, in its title, body, or one of its tags, without distinction of letter case. This search method retrieved a total of 131,620 questions. To verify that the retrieved questions are RN-related, we selected a random representative sample (383 questions, based on a 95% confidence interval). Afterward, following the guidelines of Ralph et al. [51], two authors independently reviewed each selected question to determine its relevance to RN. Then, the two authors compared their results in a joint meeting in which they calculated Cohen's kappa coefficient to measure the inter-rater agreement. The resultant Cohen's kappa coefficient was 1.00, indicating perfect agreement between the two authors [52]. Moreover, the review confirmed that all selected questions are RN-related. As a result, all of the 131,620 retrieved questions were utilized to answer the research questions of the study.

5. Results

5.1. RQ1: How Does the Interest Toward RN on SO Evolve over Time?

5.1.1. Approach. To understand how RN questions evolved over time on SO, we utilized a number of proxy measures that were utilized by previous, related studies [2, 50]. In particular, we measured the number of RN-related questions posted each year and the number of unique users who posted an RN-related question in each year. Calculating the yearly total number of RN-related questions can provide an insight into SO users' general interest in RN [2, 50]. Calculating the number of unique users who asked at least one RN-related question can provide an insight into whether RN-related questions are asked by a large number of users or by frequent, few users [2, 50]. We underscore that each user is counted once in the number of unique users; if a user asked multiple RN-related questions in 1 year or many years, we only count the user in the first year they posted the question [2, 50].

5.1.2. Results. The total number of RN-related questions that are posted yearly on SO was calculated and compared. Figure 2 presents the yearly distribution of RN-related questions posted on SO. As the figure depicts, RN was present in SO since the framework very early years. Though the number of questions posted on each year prior to 2015 (i.e., the year in which the framework was officially released) is very small, the number of questions has been steadily increasing since then throughout the years, with a drop in 2021.
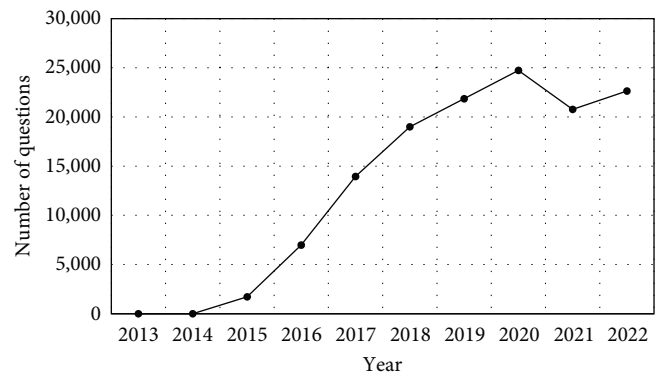

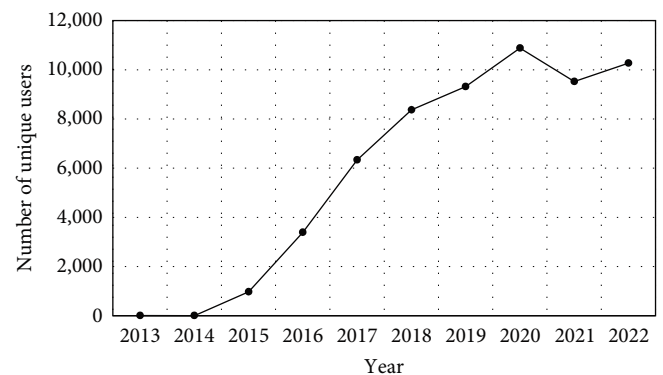FIGURE 2: Annual distribution of RN-related questions.

FIGURE 3: Annual distribution of new users who post RN-related questions.

The annual distribution of SO unique users who asked RN-related questions followed a similar trend, as depicted in Figure 3. As the figure presents, the number of new users who ask an RN-related question was pretty small prior to 2015. The number of new users has been steadily increasing since then, with a drop in 2021.

RN-related questions are present on SO since the framework early years. The annual number of questions and the annual number of unique users who asked RN-related questions have been generally increasing through the years, with drops in 2021.

5.2. RQ2: What RN-Related Topics Are Discussed on SO?

5.2.1. Approach. To identify RN topics discussed by SO users, we modeled the set of identified questions using LDA, as LDA has proved its effectiveness in topic modeling and has been successfully utilized within the software engineering field [28, 29, 53–57]. Particularly, we followed the approach illustrated in Figure 4. Initially, we preprocessed the acquired set of 131,620 RN-related questions to eliminate any noise that may impact the quality of the LDA model [39]. The preprocessing step involved removing HTML elements, HTML tags, and URLs. Afterward, the NLTK corpus was used to remove all stop words (e.g., the and is) [58]. Then, we utilized spaCY Lemmatizer to reduce words to their root form, such as converting "creating" to "create." After
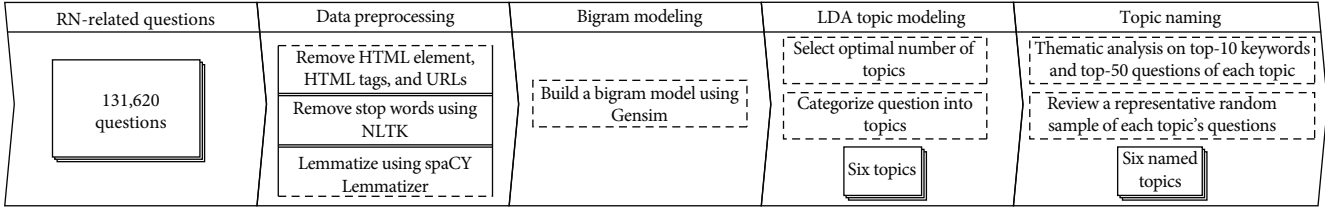
RN-related questions	Data preprocessing	Bigram modeling	LDA topic modeling	Topic naming
131,620 questions	Remove HTML element, HTML tags, and URLs Remove stop words using NLTK Lemmatize using spaCY Lemmatizer	Build a bigram model using Gensim	Select optimal number of topics Categorize question into topics Six topics	Thematic analysis on top-10 keywords and top-50 questions of each topic Review a representative random sample of each topic's questions Six named topics

FIGURE 4: Overview of topic identification approach.

TABLE 1: RN-related topics, associated top-10 keywords, and frequency.

Topic name	Keywords	Frequency
Layout and navigation	Screen, button, view, navigation, click, text, item, list, add, display	27,844
Components	Component, function, state, render, update, object, array, pass, call, change	24,645
Android issues	Error, file, build, android, version, package, fail, module, node_module, import	22,792
iOS issues	App, work, issue, ios, device, expo, open, android, test, link	20,303
Authentication and authorization	User, datum, create, api, store, log, server, message, firebase, request	19,844
Permissions and utilities	Image, time, problem, load, notification, event, video, background, implement, idea	16,192

completing the aforementioned preprocessing steps, we built a bigram model using Gensim, as bigram models have been proven to be successful in improving the quality of textual processing [30, 59]. Then, we used the LDA to identify topics that are discussed within the acquired set of RN-related questions. To identify the optimal number of topics, we conducted multiple preliminary experiments, following the approach of related, previous studies [2, 28, 30, 36, 37, 50, 59, 60]. We inputted 2 as the number of topics in the first experiment, and then we incremented the number of topics by 2 in each following experiment. We recorded the resulting coherence score of each experiment run. The coherence score is a quality measure of the resulting LDA model, where higher scores imply better quality, although it does not provide an absolute guarantee [2, 36, 59]. The experiments were stopped once passed 50 topics, as the coherence score drastically decreased. The highest coherence score was 0.62, which implies that RN-related questions can be categorized into six topics.

To determine the names of the identified topics, we conducted a thematic analysis [42]. Specifically, we utilized an inductive approach to name the set of identified topics. In this approach, no predefined set of topic names is given. Instead, the name of each topic is generated during the analysis. Therefore, two authors reviewed the top-10 keywords and the most-50-relevant questions (i.e., questions that have the 50 highest probability of belonging to a topic) of each identified topic to assign it a name. Subsequently, to verify the suitability of the assigned name to each topic, all three authors reviewed a representative, random sample of its questions [2, 30, 36, 37, 59].

5.2.2. Results. The previous approach revealed that RN-related questions on SO can be categorized into six topics. Table 1 presents the topics, their associated top-10 keywords, and frequency. Below, we provide a detailed description of each topic, sorted based on decreasing frequency.

(1) Layout and Navigation. Layout refers to the arrangement and organization of visual elements, such as images and texts, within an application [61]. RN utilizes a flex-box-based layout system that aims to provide a consistent responsive layout on different screen sizes by automatically positioning and sizing elements [2, 10, 43]. Navigation refers to the ability to move between views and screens within an application [62]. RN provides a set of libraries that aims to facilitate the navigation of applications, such as RN Navigation [34] and React Router [63]. Layout and navigation can be generally considered to be closely related, as the layout of a screen determines an application's navigation flow. Therefore, it is crucial to design the layout and navigation of an application in a cohesive manner to attain a smooth and efficient user experience. A total of 27,844 questions were identified to be belonging to the layout and navigation topic. Users asked about several issues related to layout and navigation, including nested navigation and setting up the navigation between screens. "React Native navigating between two screens back and forth," "How to navigate between different nested stacks in React Navigation without changing the state of bottom navigator," and "React Navigation switching background colors and styling StackNavigator" are examples of questions within this topic.

(2) Components. The topic of components was found to be the second largest topic, with a total of 24,645 questions. Questions in this topic inquire about all component types, including class, function, core, and custom components. Moreover, users asked about communication among components in addition to converting from one component type to another type. Examples of questions within this topic are: "How to pass multiple functions from a Class component to a Function component?", "How to pass data from a Function component to a Class component?", and "Converting Class components to Function components and using hooks."

(3) Android Issues. The Android issues topic includes 22,792 questions that are focused on RN issues related to the Android platform. Questions on this topic include compatibility issues with setting up the development environment, issues with building and compilation, issues resulting

from Android or RN updates, issues within the creation of Android Application Packages (APKs), and more. "Gradle build unsuccessful in android after Updating to 3.0.1," "react-native-gesture-handler' error when release APK in React Native," and "Update Expo SDK from 44 to 45, 46 or 47 not working" are examples of questions in this topic.

(4) *iOS Issues.* A total of 20,303 RN-related questions were found to be pertaining to iOS issues. Similar to the Android issues topic, the iOS topic contains questions that are related to development tools and environment setup issues; building and compilation errors; issues resulting from tools, iOS, or RN updates; and issues when deploying an iOS application to the Apple App Store. "Updated XCode to XCode 12.2/React Native 0.63.2 Build Fails" and "Unable to deploy my React Native app to app store (iOS)" are examples of such questions. Moreover, users asked about how to achieve a goal using RN for an iOS application and explained how will they typically achieve such a goal in Android. They have also reported on an issue that they faced within an iOS application but not within the Android version, such as "Not able to open Attachments in Salesforce Lightning Chatter page iOS WebView" and "How can a native code in Android be converted for Windows and ios using React Native."

(5) *Authentication and Authorization.* Authentication refers to verifying the identity of a user, and authorization refers to granting or restricting access to services or resources based on a user's identity [64]. A total of 19,844 RN-related questions were found to be related to the authentication and authorization topic. Questions on this topic include inquiries on how to build highly secure applications, how to fix faced errors, and what best practices can be applied in terms of authentication and authorization. Examples of such questions are "How to build a highly secure End to End Encryption React Native messaging app," "apple-signin-auth node js React Native error: invalid id token public key id," and "Authentication in react-native: best approaches." Questions also include asking about specific authentication and authorization solutions, where OAuth, Firebase, Firestore, and JSON Web Tokens (JWTs) are among the commonly discussed ones, such as "How to authenticate in Firebase by sending Otp to users via Email and "How to add JWT auth and RSA private or public key in React Native keychain?". Questions also include how to utilize secure storage solutions, such as Keychain or Secure Store, in order to securely store users' login credentials. "Ionic: I want storing authentication token only in an encrypted form, using a secure encryption key from the Android Keystore and the iOS keychain" and "Async Storage/Secure Store help for React Native" are examples of such questions.

(6) *Permissions and Utilities.* Permissions and utilities are two vital concepts that relate to accessing devices' functionalities and features. Permission refers to a user granting specific device features, such as the device camera or location services [65, 66]. These permissions can be granted or denied by a user and, typically, requested by a mobile application when needed. Utilities refer to helper functions or modules that provide common functionalities with an aim to ease development and make it more effective [67]. In the context of RN, permission and utilities are closely related in the sense that some utilities require permission to function properly. For example, the CameraRoll utility in RN allows an application to access a camera roll and display images in the application, and the utility requires a permission from the user in order to achieve that. The results revealed that 16,192 questions are related to permissions and utilities. Questions in this topic asked about a wide set of permissions and utilities, such as "How can I create an image file and 'Save to Camera Roll'," "React Native geolocation permissions callback," and "I'm building an app using the React Native with the expo in which I need the permission for the image gallery on android."

RN-related questions on SO revolve around six topics, which are as follows, ordered by decreasing frequency: layout and navigation, components, Android issues, iOS issues, authentication and authorization, and permissions and utilities.

5.3. RQ3: What Are the Characteristics of the Identified Topics, in Terms of Popularity and Difficulty, and How Are Popularity and Difficulty Related?

5.3.1. Approach. To estimate the popularity and difficulty of the topics, we adopted a number of proxy measures that were utilized in previous, related studies [2, 30, 31, 36, 37, 40, 48].

Mainly, we estimated the popularity of a topic by calculating the average number of views and score counts of its questions received [2, 30, 36, 37, 40, 50, 59]. The average number of views can serve as an indicator of the interest of both unregistered and registered users toward a topic [2]. This is particularly crucial, as it was found that there are more unregistered users than registered ones on SO [68]. Additionally, the score of a question is computed by adding the upvotes and downvotes it has received from SO users, which can serve as an indicator of how appealing the topic is to registered users on the website [30].

Afterward, to identify the most popular topic, we utilized the approach of Uddin et al. [37]. Specifically, We normalized the two popularity proxy measures by dividing the value of each measure by the average of its values across all the six identified topics. Then, for each topic, we calculated the average of the two normalized popularity proxy measures to generate a single measure that was utilized to sort the topics based on popularity. Particularly, we used the following formulas:

$$\text{View}N_i = \frac{\text{View}_i \times 6}{\sum_{j=1}^{6} \text{View}_j} \tag{1}$$

$$\text{Score}N_i = \frac{\text{Score}_i \times 6}{\sum_{j=1}^{6} \text{Score}_j} \tag{2}$$

$$\text{Fused}P_i = \frac{\text{View}N_i + \text{Score}N_i}{2} \tag{3}$$

where i denotes a specific topic.

TABLE 2: Popularity of identified topics.

Topic name	FusedP	AVG (views)	AVG (scores)
Layout and navigation	1.47	557	0.4
Components	0.82	338	0.2
Android issues	1.37	481	0.4
iOS issues	0.97	323	0.3
Authentication and authorization	0.68	242	0.2
Permissions and utilities	0.66	222	0.2

TABLE 3: Difficulty of identified topics.

Topic name	FusedD	%W/o acc answer	Med time (s)
Layout and navigation	0.79	62.1	5,802
Components	0.53	54.3	2,189
Android issues	1.42	70.9	15,815
iOS issues	1.60	72.8	18,860
Authentication and authorization	0.76	65.1	4,845
Permissions and utilities	0.86	67.4	6,363

The difficulty of a topic was also estimated by calculating two proxy measures. Mainly, we calculated the percentage of questions that did not receive an accepted answer and the median time elapsed for questions to receive an accepted answer within each topic [2, 30, 36, 37, 40, 50, 59]. The first proxy measure was utilized as the lack of an accepted answer may indicate that the questioner did not find any answer to be satisfactory and that the questioner is still struggling with an issue that they require assistance with from SO users [2, 48]. The second proxy measure was utilized due to the fact that the success of a crowd-sourced platform, such as SO, is highly dependable on its users' capability to provide quick and satisfactory answers [2, 30, 31, 49]. We note that the median is typically chosen to be calculated to compensate for the possibility of skewing of the mean due to long-latency answers [2, 30, 31, 36, 48].

Similar to the approach followed to assess the popularity of topics, we have also normalized the two difficulty proxy measures. Particularly, we divided the value of each measure by its respective average across all the six identified topics. Afterward, we calculated a single measure to facilitate the sorting of topics based on their level of difficulty. This was achieved by computing the average of the two normalized difficulty proxy measures, following the approach of Uddin et al. [37]. Specifically, we used the following formulas:

$$\%\text{W/o acc answer}N_i = \frac{\%\text{W/o acc answer}_i \times 6}{\sum_{j=1}^{6} \%\text{W/o acc answer}_j} \quad (4)$$

$$\text{Med time (s)}N_i = \frac{\text{Med time (s)}_i \times 6}{\sum_{j=1}^{6} \text{Med time (s)}_j} \quad (5)$$

$$\text{Fused}D_i = \frac{\%\text{W/o acc answer}N_i + \text{Med time (s)}N_i}{2} \quad (6)$$

where i denotes a specific topic.

Lastly, we expanded the scope of our analysis by examining the relationships among each popularity and difficulty proxy measure. Particularly, we utilized the Kendall correlation test to assess the monotonic correlation between the popularity and difficulty proxy measures [69]. We decided to utilize the test due to the test tolerance for outliers and the capability of producing robust results [69, 70].

5.3.2. Results. The obtained statistics on the popularity of each identified RN-related topic are presented in Table 2, where column "FusedP" presents the calculated fused popularity measure, column "AVG(views)" presents an average number of views, and column "AVG(scores)" presents an average number of scores. As presented in Table 2, the topic of layout and navigation has the highest FusedP, and hence, it can be considered the most popular topic. The topic of Android issues has the second highest FusedP, followed by iOS issues, components, authentication and authorization, and permissions and utilities.

The layout and navigation topic can be considered the most popular identified RN-related topic.

The results of the assessment of topics' difficulty are presented in Table 3, where column "FusedD" presents fused difficulty measure, column "%W/o acc answer" presents percentages of questions without an accepted answer, and column "Med time (s)" presents median time elapsed to receive an accepted answer in seconds. As the table presents, the topic of iOS issues has the highest FusedD, followed by Android issues, permissions and utilities, layout and navigation, authentication and authorization, and components. Based on the results, it can be concluded that iOS issues are the most difficult RN-related topic, followed by Android issues.

The iOS issues topic can be considered the most difficult topic, followed by Android issues.

Moreover, the examination of the monotonic correlation between each popularity and difficulty proxy measure using

TABLE 4: Correlation coefficients of popularity and difficulty proxy measures.

	View	Score
%W/o acc	0.2	0.1
Time to acc	0.1	0.1

the Kendall rank correlation test resulted in p-values less than 0.01 in all cases, indicating statistically significant correlations. As presented in Table 4, though the resulting correlation coefficients are positive, all of the resulting coefficients are less than 0.2, indicating negligible correlations [71, 72].

There is a negligible positive correlation between the popularity and difficulty proxy measures.

6. Discussion

The results divulged that interest in RN on SO has been steadily increasing since its release, with a drop in 2021. Though the framework was officially released in 2015, discussions and interest in RN have been present on SO prior to the framework's official release and have been continuously growing since then, with a peak in 2020 and a drop in 2021. Moreover, the results revealed that RN-related questions on SO mainly focus on six topics. The presence of some of the topics was expected due to them being related to common challenges in mobile application development and software development in general, such as authentication and authorization [73, 74]. Other topics were also expected to appear in RN-related questions, such as layout and navigation, as they are commonly considered to be challenging in mobile application development [2, 10, 43]. On the contrary, it was unexpected that SO users' discussions that revolve around RN include some of the topics, such as components, as the topic has been thoroughly explained in the official RN documentation that includes various resources, such as examples, video tutorials, etc. Additionally, the results revealed that the layout and navigation topic can be considered the most popular topic and that the topics of iOS issues and Android issues can be considered the most challenging topics. Finding that the layout and navigation topic to be considered the most popular was expected, as layout and navigation is a core challenge within mobile application development [2] due to the fact that developers need to account for a wide range of mobile devices that have varying capabilities and screen sizes [75]. Moreover, the topic appeared in previous, related studies that analyzed mobile developers' discussions [2, 10, 43]. Furthermore, finding that the topics that are related to a specific platform (i.e., iOS and Android) to be the two most challenging topics can be attributed to the limited official support for RN from these operating systems, leading to increased challenges in development and longer resolution times for any issues that arise [76]. Lastly, the examinations of potential correlations between each popularity and difficulty proxy measures have resulted in very low correlation coefficients, indicating negligible positive associations between popularity and difficulty.

7. Implications

SO has continuously demonstrated its utility in gaining insights into the status of varying subjects within the field of software engineering [2, 28]. This study identifies and analyzes RN-related questions acquired from SO, and this section summarizes how software engineering researchers, practitioners, educators, and RN contributors may utilize the findings of this study.

Software engineering researchers may find this study's results to be beneficial in guiding their future RN-related research. As an example, the topics of iOS issues and Android issues were found to be considered difficult topics, with the majority of questions lacking an accepted answer, despite RN efforts to ease the development process through providing guidelines for each platform in the official documentation. Researchers may investigate specific iOS and Android problems that developers are experiencing when using RN to research and develop new solutions or address any shortcomings within the existing ones.

Practitioners of software engineering may acquire an insight into the RN-related topics discussed on SO and the characteristics of these topics. For instance, the topic of layout and navigation was found to be the most popular topic, with a relatively high rate of accepted answers. Examining questions related to the topic may aid practitioners in being aware of the trends in layout and navigation, and they might consider leveraging these trends when constructing their applications. Moreover, having knowledge regarding the most difficult RN-related topics can aid developers realizing that these challenges are common and that some of the challenges have no known solutions. As a result, practitioners can become precautious about them and either avoid or mitigate such challenges based on other users' experiences when confronting similar problems. For instance, practitioners can anticipate facing hurdles related to iOS and Android, and based on that anticipation, they can attempt to find solutions for these problems prior to investing resources, or they can instead seek for other alternatives.

Moreover, software engineering educators can utilize the results of this study to improve their curricula on mobile development. The educators may consider including RN in their teaching, as the study found that interest toward RN is generally increasing. Educators may also consider focusing on RN-related topics that SO users are finding troublesome, such as iOS and Android issues. Incorporating such topics may aid students in expecting potential problems, understanding them, and mitigating or tackling them.

RN contributors may utilize the results of this study to identify challenges that RN users are facing and asking about, such as iOS issues. Contributors may find such knowledge beneficial when they attempt to remove such challenges or provide guidelines to mitigate these challenges. Furthermore, as the study found that the users of SO are inquiring about topics that have been thoroughly explained in RN documentation, such as components, contributors may identify what is unclear or missing, if any, within the current documentation to better explain it or add it. They may also leverage SO

to promote the documentation by answering related questions and direct questioners to the parts of the documentation that address their questions.

8. Threats to Validity

This section presents potential threats to this study's validity. External, construct, and reliability threats and their mitigation strategies, if applicable, are summarized below.

8.1. External Validity. The external validity of a study refers to the extent to which the results of the study can be generalized and applied to other contexts [77–79]. This study has specifically focused on RN-related questions on SO, aiming to gain a deeper understanding of these questions on the website. Therefore, the findings of this study cannot be generalized to RN discussions on other platforms, as there is a possibility that there exist other RN-related topics discussed on other platforms. Additionally, though we strove to include all RN-related questions on SO by utilizing a search method that leverages both tag-based and content-based filtering, there is still a possibility that we failed to include some related questions, as users may have used different terms to reference RN. Therefore, the results of this study are limited to questions identified using the described search method.

8.2. Construct Validity. Construct validity refers to the extent to which the measures used in a study accurately measure what is intended to be measured [77–79]. Two potential threats to the construct validity of this study are identified. First, the study measures interest, popularity, and difficulty of RN-related topics solely through proxy measures. Although the utilized proxy measures were commonly employed in previous, related studies [2, 30, 31, 36, 40, 48, 73], the findings should only be interpreted within the context of these proxy measures. The second threat to the construct validity of this study is the usage of statistical measures and tests. We only utilized common statistical measures, such as percentages and median; the Cohen's kappa coefficient to measure inter-rater agreement; and the Kendall correlation test to examine the relationship between each popularity and difficulty proxy measure. These statistical measures and tests are commonly used in the contexts in that we utilized them in, and the utilization of the majority of these tests and measures is recommended by the empirical standards for software engineering research [51, 80].

8.3. Reliability. The reliability of a study refers to the degree to which the study and its results can be replicated by other researchers. In other words, future researchers should be able to reach the same conclusions and findings following the same methodologies followed in studies [78, 79]. One potential threat to this study's reliability is the usage of human judgment in assigning names to the identified RN-related topics, which can introduce subjectivity. To mitigate this threat, we conducted a thematic analysis that involved reviewing the top-10 keywords, top-50 questions, and a representative random sample of each topic to ensure the suitability of the name assigned to each topic. Despite these efforts, we acknowledge that our naming approach still suffers from subjectivity and that other human-based reviews may result in

different names. However, we note that this subjectivity could not be avoided, as the LDA only groups questions based on their cooccurrence appearance frequency without providing names, and there is no automated approach to name the resulting topics [2, 28].

9. Conclusion

RN is an open-source framework that allows developers to build native mobile applications for different platforms using a single codebase. The framework has been widely used by developers to create high-performing cross-platform mobile applications. With an objective of gaining a better understanding of the current questions revolving around RN on SO, this study analyzes SO RN-related questions.

The study involved the collection and analysis of 131,620 SO RN-related questions. The analysis revealed that interest toward RN on SO has been generally increasing, with a drop in 2021. Additionally, modeling the identified set of questions using LDA revealed that RN-related questions revolve around six topics, which are sorted in decreasing frequency: layout and navigation, components, Android issues, iOS issues, authentication and authorization, and permissions and utilities. Moreover, the utilization of proxy measures revealed that the identified topics vary in terms of their popularity and difficulty levels, where the topic of layout and navigation can be considered the most popular topic and topics that are related to a specific platform (i.e., iOS and Android) can be considered the most difficult topics.

This study provides an insight into the stance of RN-related questions on SO. The study highlights and characterizes topics that the website's users discussed in the context of RN. Software engineering practitioners, researchers, educators, and RN contributors can utilize this study as a guide to focus their future RN-related efforts.

Data Availability

The data that support the findings of this study are available from the corresponding author upon reasonable request.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

References

[1] A. Ahmad, K. Li, C. Feng, S. M. Asim, A. Yousif, and S. Ge, "An empirical study of investigating mobile applications development challenges," *IEEE Access*, vol. 6, pp. 17711–17728, 2018.

[2] C. Rosen and E. Shihab, "What are mobile developers asking about? A large scale study using stack overflow," *Empirical Software Engineering*, vol. 21, no. 3, pp. 1192–1223, 2016.

[3] C. Tam, D. Santos, and T. Oliveira, "Exploring the influential factors of continuance intention to use mobile apps: extending the expectation confirmation model," *Information Systems Frontiers*, vol. 22, no. 1, pp. 243–257, 2020.

[4] K. Hirsh-Pasek, J. M. Zosh, R. M. Golinkoff, J. H. Gray, M. B. Robb, and J. Kaufman, "Putting education in "Educational" apps: lessons from the science of learning," *Psychological Science in the Public Interest*, vol. 16, no. 1, pp. 3–34, 2015.

[5] F. Nausheen and S. H. Begum, "Healthcare IoT: benefits, vulnerabilities and solutions," in *2018 2nd International Conference on Inventive Systems and Control (ICISC)*, pp. 517–522, IEEE, Coimbatore, India, 2018.

[6] H. Taherdoost, "Fintech: emerging trends and the future of finance," in *Financial Technologies and DeFi*, Financial Innovation and Technology, pp. 29–39, Springer, Cham, 2023.

[7] M.-E. Angelaki, T. Karvounidis, and C. Douligeris, "Mobile applications and projects for sustainable urban public transportation: a selective review," in *24th Pan-Hellenic Conference on Informatics*, pp. 156–161, ACM, 2020.

[8] L. Ceci, "Mobile app usage-statistics & facts," 2021, [Online]. Available: https://www.statista.com/topics/1002/mobile-app-usage/#dossierKeyfigures.

[9] R. Minelli and M. Lanza, "Software analytics for mobile applications–insights & lessons learned," in *2013 17th European Conference on Software Maintenance and Reengineering*, pp. 144–153, IEEE, Genova, Italy, 2013.

[10] M. Linares-Vásquez, B. Dit, and D. Poshyvanyk, "An exploratory analysis of mobile development issues using stack overflow," in *2013 10th Working Conference on Mining Software Repositories (MSR)*, pp. 93–96, IEEE, San Francisco, CA, USA, 2013.

[11] M. E. Joorabchi, A. Mesbah, and P. Kruchten, "Real challenges in mobile app development," in *2013 ACM/IEEE International Symposium on Empirical Software Engineering and Measurement*, pp. 15–24, IEEE, Baltimore, MD, USA, 2013.

[12] K. Moran, C. Watson, J. Hoskins, G. Purnell, and D. Poshyvanyk, "Detecting and summarizing GUI changes in evolving mobile apps," in *Proceedings of the 33rd ACM/IEEE International Conference on Automated Software Engineering*, pp. 543–553, IEEE, 2018.

[13] K. Moran, C. Bernal-Cárdenas, M. Linares-Vásquez, and D. Poshyvanyk, "Overcoming language dichotomies: toward effective program comprehension for mobile app development," in *Proceedings of the 26th Conference on Program Comprehension*, pp. 7–18, ACM, 2018.

[14] A. I. Wasserman, "Software engineering issues for mobile application development," in *Proceedings of the FSE/SDP workshop on Future of software engineering research*, pp. 397–400, ACM, 2010.

[15] X. Jia, A. Ebone, and Y. Tan, "A performance evaluation of cross-platform mobile application development approaches," in *2018 IEEE/ACM 5th International Conference on Mobile Software Engineering and Systems (MOBILESoft)*, pp. 92-93, IEEE, Gothenburg, Sweden, 2018.

[16] E. Windmill, *Flutter in Action*, Simon and Schuster, 2020.

[17] M. Martinez and S. Lecomte, "Towards the quality improvement of cross-platform mobile applications," in *2017 IEEE/ACM 4th International Conference on Mobile Software Engineering and Systems (MOBILESoft)*, pp. 184–188, IEEE, Buenos Aires, Argentina, 2017.

[18] C. R. Raj and S. B. Tolety, "A study on approaches to build cross-platform mobile applications and criteria to select appropriate approach," in *2012 Annual IEEE India Conference (INDICON)*, pp. 625–629, IEEE, Kochi, India, 2012.

[19] A. Ribeiro and A. R. da Silva, "Survey on cross-platforms and languages for mobile apps," in *2012 Eighth International Conference on the Quality of Information and Communications Technology*, pp. 255–260, IEEE, Lisbon, Portugal, 2012.

[20] M. Latif, Y. Lakhrissi, E. H. Nfaoui, and N. Es-Sbai, "Cross platform approach for mobile application development: a survey," in *2016 International Conference on Information Technology for Organizations Development (IT4OD)*, pp. 1–5, IEEE, Fez, Morocco, 2016.

[21] A. Kuttig, *Professional React Native: Expert Techniques and Solutions for Building High-Quality, Cross-Platform, Production-Ready Apps*, Packt Publishing, 2022, [Online]. Available: https://books.google.com.sa/books?id=pXuUEAAAQBAJ.

[22] B. Eisenman, *Learning React Native: Building Native Mobile Apps with JavaScript*, "O'Reilly Media, Inc.", 2015.

[23] A. Desmurs-Linczewska, *Simplifying State Management in React Native: Master State Management from Hooks and Context through to Redux, MobX, XState, Jotai and React Query*, Packt Publishing, Birmingham, England, 2023.

[24] E. Masiello and J. Friedmann, *Mastering React Native*, Packt Publishing Ltd., 2017.

[25] A. Boduch, *React and React Native*, Packt Publishing Ltd., 2017.

[26] R. Native, "React Native learn once, write anywhere," 2022.

[27] K. Bajaj, K. Pattabiraman, and A. Mesbah, "Mining questions asked by web developers," in *Proceedings of the 11th Working Conference on Mining Software Repositories*, pp. 112–121, ACM, 2014.

[28] C. C. Silva, M. Galster, and F. Gilson, "Topic modeling in software engineering research," *Empirical Software Engineering*, vol. 26, no. 6, pp. 1–62, 2021.

[29] S. Ahmed and M. Bagherzadeh, "What do concurrency developers ask about? A large-scale study using stack overflow," in *Proceedings of the 12th ACM/IEEE International Symposium on Empirical Software Engineering and Measurement*, pp. 1–10, ACM, 2018.

[30] M. U. Haque, L. H. Iwaya, and M. A. Babar, "Challenges in docker development: a large-scale study using stack overflow," in *Proceedings of the 14th ACM/IEEE International Symposium on Empirical Software Engineering and Measurement (ESEM)*, pp. 1–11, ACM, 2020.

[31] A. Abdellatif, D. Costa, K. Badran, R. Abdalkareem, and E. Shihab, "Challenges in chatbot development: a study of stack overflow posts," in *Proceedings of the 17th International Conference on Mining Software Repositories*, pp. 174–185, ACM, 2020.

[32] D. M. Blei, A. Y. Ng, and M. I. Jordan, "Latent Dirichlet allocation," *The Journal of Machine Learning Research*, vol. 3, pp. 993–1022, 2003.

[33] M. Grzesiukiewicz, *Hands-On Design Patterns with React Native: Proven Techniques and Patterns for Efficient Native Mobile Development with JavaScript*, Packt Publishing, 2018.

[34] N. Dabit, *React Native in Action*, Manning Publications, 2019.

[35] A. B. Kuttig, *Professional React Native: Expert Techniques and Solutions for Building High-Quality, Cross-Platform, Production-Ready Apps*, Packt Publishing, 2022.

[36] R. Alfayez, Y. Ding, R. Winn, and G. Alfayez, "What is discussed about software engineering ethics on stack exchange (Q&A) websites? A case study," in *2022 IEEE/ACIS 20th*

International Conference on Software Engineering Research, Management and Applications (SERA), pp. 106–111, IEEE, Las Vegas, NV, USA, 2022.

[37] G. Uddin, F. Sabir, Y.-G. Guéhéneuc, O. Alam, and F. Khomh, "An empirical study of IoT topics in IoT developer discussions on stack overflow," *Empirical Software Engineering*, vol. 26, Article ID 121, 2021.

[38] H. Jelodar, Y. Wang, C. Yuan et al., "Latent Dirichlet allocation (LDA) and topic modeling: models, applications, a survey," *Multimedia Tools and Applications*, vol. 78, pp. 15169–15211, 2019.

[39] A. Barua, S. W. Thomas, and A. E. Hassan, "What are developers talking about? An analysis of topics and trends in stack overflow," *Empirical Software Engineering*, vol. 19, no. 3, pp. 619–654, 2014.

[40] J. Han, E. Shihab, Z. Wan, S. Deng, and X. Xia, "What do programmers discuss about deep learning frameworks," *Empirical Software Engineering*, vol. 25, no. 4, pp. 2694–2747, 2020.

[41] S. Fincher and J. Tenenberg, "Making sense of card sorting data," *Expert Systems*, vol. 22, no. 3, pp. 89–93, 2005.

[42] D. S. Cruzes and T. Dyba, "Recommended steps for thematic synthesis in software engineering," in *2011 International Symposium on Empirical Software Engineering and Measurement*, pp. 275–284, IEEE, Banff, AB, Canada, 2011.

[43] S. Beyer and M. Pinzger, "A manual categorization of Android app development issues on stack overflow," in *2014 IEEE International Conference on Software Maintenance and Evolution*, pp. 531–535, IEEE, Victoria, BC, Canada, 2014.

[44] V. Oliveira, L. Teixeira, and F. Ebert, "On the adoption of Kotlin on Android development: a triangulation study," in *2020 IEEE 27th International Conference on Software Analysis, Evolution and Reengineering (SANER)*, pp. 206–216, IEEE, London, ON, Canada, 2020.

[45] J. Zou, L. Xu, M. Yang, X. Zhang, and D. Yang, "Towards comprehending the non-functional requirements through developers' eyes: an exploration of stack overflow using topic analysis," *Information and Software Technology*, vol. 84, pp. 19–32, 2017.

[46] A. Bandeira, C. A. Medeiros, M. Paixao, and P. H. Maia, "We need to talk about microservices: an analysis from the discussions on stackoverflow," in *2019 IEEE/ACM 16th International Conference on Mining Software Repositories (MSR)*, pp. 255–259, IEEE, Montreal, QC, Canada, 2019.

[47] M. Bagherzadeh and R. Khatchadourian, "Going big: a large-scale study on what big data developers ask," in *Proceedings of the 2019 27th ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering*, pp. 432–442, ACM, 2019.

[48] M. Alshangiti, H. Sapkota, P. K. Murukannaiah, X. Liu, and Q. Yu, "Why is developing machine learning applications challenging? A study on stack overflow posts," in *2019 ACM/IEEE International Symposium on Empirical Software Engineering and Measurement (ESEM)*, pp. 1–11, IEEE, Porto de Galinhas, Brazil, 2019.

[49] R. Alfayez, Y. Ding, R. Winn, G. Alfayez, C. Harman, and B. Boehm, "What is asked about technical debt (TD) on stack exchange question-and-answer (Q&A) websites? An observational study," *Empirical Software Engineering*, vol. 28, no. 2, Article ID 35, 2023.

[50] A. Peruma, S. Simmons, E. A. AlOmar, C. D. Newman, M. W. Mkaouer, and A. Ouni, "How do I refactor this? An empirical study on refactoring trends and topics in stack overflow," *Empirical Software Engineering*, vol. 27, no. 1, Article ID 11, 2022.

[51] P. Ralph, N. bin Ali, S. Baltes et al., "Empirical standards for software engineering research," arXiv preprint arXiv: 2010.03525, 2020.

[52] J. R. Landis and G. G. Koch, "The measurement of observer agreement for categorical data," *Biometrics*, vol. 33, no. 1, pp. 159–174, 1977.

[53] V. Johri and S. Bansal, "Identifying trends in technologies and programming languages using topic modeling," in *2018 IEEE 12th International Conference on Semantic Computing (ICSC)*, pp. 391–396, IEEE, Laguna Hills, CA, USA, 2018.

[54] K. J. Gülle, N. Ford, P. Ebel, F. Brokhausen, and A. Vogelsang, "Topic modeling on user stories using word mover's distance," in *2020 IEEE Seventh International Workshop on Artificial Intelligence for Requirements Engineering (AIRE)*, pp. 52–60, IEEE, 2020.

[55] H. U. Asuncion, A. U. Asuncion, and R. N. Taylor, "Software traceability with topic modeling," in *Proceedings of the 32nd ACM/IEEE international conference on Software Engineering-Volume 1*, pp. 95–104, ACM, 2010.

[56] S. W. Thomas, "Mining software repositories using topic models," in *2011 33rd International Conference on Software Engineering (ICSE)*, pp. 1138-1139, IEEE, Honolulu, HI, USA, 2011.

[57] A. Panichella, B. Dit, R. Oliveto, M. Di Penta, D. Poshynanyk, and A. De Lucia, "How to effectively use topic models for software engineering tasks? An approach based on genetic algorithms," in *2013 35th International conference on software engineering (ICSE)*, pp. 522–531, IEEE, San Francisco, CA, 2013.

[58] S. Bird, E. Klein, and E. Loper, *Natural Language Processing with Python: Analyzing Text with the Natural Language Toolkit*, "O'Reilly Media, Inc.", 2009.

[59] K. Mahmood, G. Rasool, F. Sabir, and A. Athar, "An empirical study of web services topics in web developer discussions on stack overflow," *IEEE Access*, vol. 11, pp. 9627–9655, 2023.

[60] R. Alfayez, R. Winn, Y. Ding, G. Alfayez, and B. Boehm, "Technical debt (TD) through the lens of Twitter: a survey," *Journal of Software: Evolution and Process*, Article ID e2547, 2023.

[61] A. Majooni, M. Masood, and A. Akhavan, "An eye-tracking study on the effect of infographic structures on viewer's comprehension and cognitive load," *Information Visualization*, vol. 17, no. 3, pp. 257–266, 2018.

[62] N. Jailani, Z. Abdullah, M. A. Bakar, and H. R. Haron, "Usability guidelines for developing mobile application in the construction industry," in *2015 International Conference on Electrical Engineering and Informatics (ICEEI)*, pp. 411–416, IEEE, Denpasar, Indonesia, 2015.

[63] D. B. Duldulao and R. J. L. Cabagnot, "Navigating react router," in *Practical Enterprise React*, pp. 55–90, Springer, 2021.

[64] M. Wazid, S. Zeadally, and A. K. Das, "Mobile banking: evolution and threats: malware threats and security solutions," *IEEE Consumer Electronics Magazine*, vol. 8, no. 2, pp. 56–60, 2019.

[65] E. Holmes and T. Bray, *Getting Started with React Native, ser. Community Experience Distilled*, Packt Publishing, 2015.

[66] G. L. Scoccia, M. Autili, and P. Inverardi, "A self-configuring and adaptive privacy-aware permission system for Android apps," in *2020 IEEE International Conference on Autonomic Computing and Self-Organizing Systems (ACSOS)*, pp. 38–47, IEEE, Washington, DC, USA, 2020.

[67] A. Desmurs-Linczewska, *Simplifying State Management in React Native: Master State Management From Hooks and Context through to Redux, MobX, XState, Jotai and React Query*, Packt Publishing, 2023.

[68] L. Mamykina, B. Manoim, M. Mittal, G. Hripcsak, and B. Hartmann, "Design lessons from the fastest Q&A site in the west," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pp. 2857–2866, ACM, 2011.

[69] H. Abdi, "The Kendall rank correlation coefficient," *Encyclopedia of Measurement and Statistics*, vol. 2, pp. 508–510, 2007.

[70] M.-T. Puth, M. Neuhäuser, and G. D. Ruxton, "Effective use of Spearman's and Kendall's correlation coefficients for association between two measured traits," *Animal Behaviour*, vol. 102, pp. 77–84, 2015.

[71] P. Schober, C. Boer, and L. A. Schwarte, "Correlation coefficients: appropriate use and interpretation," *Anesthesia & Analgesia*, vol. 126, no. 5, pp. 1763–1768, 2018.

[72] M. Hubert and P. Rousseeuw, *International Encyclopedia of Statistical Science*, Springer-Verlag, 2010.

[73] X.-L. Yang, D. Lo, X. Xia, Z.-Y. Wan, and J.-L. Sun, "What security questions do developers ask? A large-scale study of stack overflow posts," *Journal of Computer Science and Technology*, vol. 31, pp. 910–924, 2016.

[74] N. Davis, "Developing secure software," *Software Tech News*, vol. 8, no. 2, pp. 3–7, 2005.

[75] A. Aldayel and K. Alnafjan, "Challenges and best practices for mobile application development," in *Proceedings of the International Conference on Compute and Data Analysis*, pp. 41–48, ACM, 2017.

[76] P. Nawrocki, K. Wrona, M. Marczak, and B. Sniezynski, "A comparison of native and cross-platform frameworks for mobile applications," *Computer*, vol. 54, no. 3, pp. 18–27, 2021.

[77] B. A. Kitchenham, D. Budgen, and P. Brereton, *Evidence-Based Software Engineering and Systematic Reviews*, CRC Press, 2015.

[78] P. Runeson, M. Host, A. Rainer, and B. Regnell, *Case Study Research in Software Engineering: Guidelines and Examples*, John Wiley & Sons, 2012.

[79] R. K. Yin, *Case Study Research: Design and Methods*, Sage, 2009.

[80] N. Gisev, J. S. Bell, and T. F. Chen, "Interrater agreement and interrater reliability: key concepts, approaches, and applications," *Research in Social and Administrative Pharmacy*, vol. 9, no. 3, pp. 330–338, 2013.